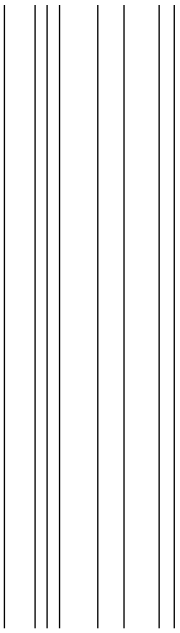


2013 | Faculteit Wetenschappen

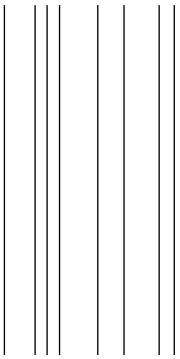
DOCTORAATSPROEFSCHRIFT

Analyzing the Predictive Performance of Scheduler Process Models of Rule-based Activity-based Models

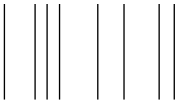
Proefschrift voorgelegd tot het behalen van de graad van doctor in de wetenschappen, te verdedigen door:

George Sammour

Promotor: prof. dr. Koen Vanhoof
Copromotor: prof. dr. Tom Bellemans



D/2013/2451/4



universiteit hasselt
KNOWLEDGE IN ACTION

Acknowledgement

First of all I would like to thank Hasselt University and my colleagues at the Transportation Research Institute (IMOB) for their stimulating support. This research work would not have been possible without their support. I would like to thank all the people who have helped and inspired me during my doctoral study.

I wish to express my sincere gratitude to my promotor Prof. Dr. Koen Vanhoof. I consider it an honor working under your supervision. I have gained very much from your professional experience and knowledge in doing research. Your profound advices, positive criticism, and kind patience have assisted me in times I needed it the most. I will never forget your response to one of my emails when I discussed some agonized issues related to my research, and I quote “George, if you are worried that much, then that is a healthy sign”.

I would like to gratefully acknowledge the enthusiastic supervision of Prof. Dr. Tom Bellemans. Your skeptic and stressful critiques have been relevant and helpful. Without your feedback, this research would not have been a success. I would like to thank my PhD committee members Prof. Dr. Geert Wets and Prof. Dr. Davy Janssens for their valuable comments, discussions, and thorough guidance during my quarterly PhD progress meetings. I wish also to express my sincere gratitude to Prof. Dr. Benoit Depaire for his endless support for helping me comprehend many aspects of data mining.

I cannot find words to express my gratitude for my beloved wife, Sawsan. Thank you for bearing and holding responsibility of our children for more than three years. Your kindness, endless patience, and love have been and will always be the source of success in my life. My children, Fadi, Sandy, and Sulaiman deserve also my appreciation for the understanding that they have shown through a long period during which I have not been able to be with them. I would also like to

thank my parents and sisters. They were always supporting me and encouraging me with their best wishes.

I am indebted to my many colleagues and friends who contributed to my PhD research and supported me during my stay in Belgium. In particular, I would like to thank Ali Pirdavani, Konstantinos Perraks, Dirk Roox, and Bruno Kochan. In addition, I would like to extend my gratitude to many dear friends, Hans tormans, Mario Cools, Marlise Vanhulsel, Else Hannes, and Diana Kusumastuti. Special thanks also goes to Kristel Hertogs for all her administrative and practical assistance since the first day for me in Belgium.

Last but not least I am highly indebted to Prof. Dr. Abdullah Alzoubi, former chair of the Scientific Research Deanship at Princess Sumaya University for Technology (PSUT) in Jordan. Your endless support was the main reason to pursue my PhD opportunity. I would like to thank Dr. Jeanne Shreurs for her continual support before I started my PhD. Finally, I would like to express my sincere thankfulness to my best teacher, Prof. Dr. Walid Salameh, chair of the Scientific Research Deanship at PSUT. Thank you and all members of the Jury for your thoughtful and valuable remarks.

George Sammour
January 10th, 2013

Executive Summary

Activity-based travel demand modeling systems to date can be classified into two modeling approaches Utility maximization-based econometric model systems, and Rule-based computational process model systems. This dissertation discusses three particular contributions with respect to rule-based computational process models specifically the ALBATROSS (A Learning-based Transportation Oriented Simulation System) models system. The first contribution is related to improving the predictive performance of the scheduling process models. The second contribution involves analysing performance boundaries for rule-based activity forecasting models. And the third contribution is to conduct a sensitivity analysis of the models at each decision step (decision tree models). To achieve the goals, the ALBAROSS model is integrated in the FEATHERS framework. FEATHERS (Forecasting Evolutionary Activity-Travel of Households and their Environmental RepercussionS) is an experimental framework developed to facilitate the development of modular activity-based models for transportation demand. To include ALBATROSS in FEATHERS (FEATALB), the model parameters were modified to fit the Flemish data. The ALBATROSS model and its components have been studied in details. However, some practical limitations were determined that restrained further experimentations and there was a need for new implementations. Some parts of the model were re-implemented. The implementation involved using technologies to boost the design of experiments conducted in this thesis.

There are three major factors related to improve the predictive performance of rule-based models. The first factor is to ensure that the data are of good quality i.e. obtaining better data that are used to train the models at individual decision steps. The second factor involves utilizing better classifiers at individual decision steps that constitute the scheduling process model. The third factor is to achieve a better data representation in the context of reordering the decision steps in the process model.

The methodology to achieve the first goal is obtained by training the decision models in three different approaches. First by modeling all the decision models in the process model simultaneously, using a multi-target classification method. Using a multi-target classification method eliminates the activation dependency and attributes interdependencies features and it has the lowest fitting capacity. Second by training the decision models without the attributes interdependencies. This allowed investigating the added value of this feature in the model. Third by training the models at decision steps preserving the attributes interdependencies among models (fully-informed approach) while including observed rather predicted decision outcomes in subsequent decision steps. To investigate the classification method factor, the non-informed and fully-informed approaches are examined using three classification methods, CHAID, C4.5, and Logistic regression methods.

The second contribution was related to investigating the data representation factor to improve the predictive performance of process models used in activity-based models. This is achieved by presenting three different process models, i.e. the activation dependency feature by changing the order of decision models in the process model.

The third contribution which is related to experiment the sensitivity of the models at each decision step (decision tree models). The sensitivity analysis was performed by experimenting two important factors used in the decision tree models in FEATALB. The first sensitivity factor involved identifying the ideal number of minimum cases per leaf node while training the decision tree models. In ALBATROSS this number was set to 30, however, this number was set to be used in the Dutch data. For the Flemish data, a different number might improve the performance. The second sensitivity factor is the action assignment rule used in predicting values at decision steps. ALBATROSS suggests a probabilistic action assignment rule which considers the probability of predicting a specific class. Rather than predicting a class variable according to the plurality rule. The

work reported in this thesis was conducted in within the FEATALB framework. The FEATALB framework is based on the FEATHERS framework, which currently integrates the ALBATROSS model as its core scheduling system.

The predictive performance of the approaches and models experimented was assessed at three levels. The models at the individual classifier level are validated using Confusion Matrix statistics and the Brier Score. Where at the activity pattern level, the models are validated using the Sequence Alignment Method (SAM). And at the spatial and temporal level, OD matrices and work activity start times statistics are calculated using the correlation coefficient.

The results of analyzing the factors affecting the predictive performance of activity-based models show that the attributes interdependencies feature is a critical factor. Maintaining this feature in activity-based models enhances the predictive performance. However, in this context, another factor that affects the predictive performance of process models is the relevance of the decision outcome that is added in subsequent decision models in predicting the class variable. On the other hand the results also suggest that the disposition of decision steps (activation dependency feature) or experiment other data representations within the work activity process model does not lead to significantly improve the predictive performance of the model. This is confirmed by the validation at the aggregated levels (activity pattern and Spatial and temporal levels). Hence, using the currently implemented work activity process model can achieve satisfactory work activity schedules.

Considering the results of experiment the decision tree parameters (modifying the minimum number of cases at leaf nodes) suggest that increasing the minimum number of cases at leaf nodes to more than 30 cases will result in model under-fitting. Therefore, the models are trained by decreasing the minimum number of cases at leaf nodes to 20 and 5. The experimental results show that by decreasing the number to 20 cases at leaf nodes has no significant

effect on the predictive performance of the model. In addition, decreasing the number to 5 results in increase over-fitting and thus, decreases the predictive performance of the models.

Table of Content

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction and Research Motivation

In the past few decades, various models of activity scheduling behaviour have been developed. These models are part of more comprehensive activity-based models of travel demand that predict which activities will be conducted, where, when, for how long, with whom and mode choice involved (Timmermans et al., 2002). Activity-based models of travel demand are classified into two main approaches the *simultaneous* approach and the *sequential* approach (Ettema and Timmermans, 1997). The first approach employs a simultaneous choice among a set of activity-travel patterns. Activity-travel patterns are identified by a set of attributes that are incorporated in the model. Examples of such models include CARLA (Jones et al., 1983), STARCHILD (Recker et al., 1986), and Wen and Koppelman (1999, and Wen, 1998). In the simultaneous approach, activity-travel patterns are predicted using a utility-maximisation framework. Therefore, individuals are assumed to plan their activities such that their utilities are maximized, subject to a set of constraints. Utility-maximisation models predict activity-travel patters using multinomial logit models. However, the utility-maximisation approach has been criticized that they do not always reflect the true behaviour underlying travel decisions. People tend to reason more logically in terms of heuristics based on “If-then-else” rules (Janssens et al., 2004).

The second approach (sequential models) focuses on the need for rule-based computational process models. In their most basic form, rule-based computational process models use a set of “If-then” rules. Examples of such models include AMOS (Pendyala et al., 1995, 1998), FAMOS (Pendyala, 2004) and ALBATROSS (Arentze & Timmermans, 2000). Rule-based activity-based models have proven to be more flexible than utility-maximising models (Arentze et al, 2001) and they also perform well in predicting transport choice behaviour if

an induction technique is used (Wets et al, 2000). However, computational process activity-based models are argued to lack ease of interpretation, and hard to statistically assess the decision-rules performance (Moons, 2005). As a result, even though the computational process rule-based activity-based models are developed to better reflect the behavioural characteristic underlying activity-travel decisions, such models are viewed as black boxes. Examples of such models include SCHEDULER (GÄarling et al., 1989), AMOS (Pendyala et al., 1995, 1998), and ALBATROSS (Arentze and Timmermans, 2000).

Rule-based activity-based models aim at predicting which activities are conducted, where, when, with whom, for how long, and the transportation mode. Since one of the most important applications of Artificial Intelligence (AI) is decision making, this sequence of decisions shows the convenience of applying AI techniques. AI techniques can be divided into two broad categories, knowledge representation systems and machine learning systems. Knowledge representation systems provide a configuration for capturing and representing the knowledge of a human expert in a particular domain. While, machine learning systems, such as, neural networks, Induction (classification) methods, and genetic algorithms, aims at deriving decisions or solutions by learning patterns in data. Given that in rule-based activity-based models a set of rules is adapted to predict activity schedules, machine learning methods are typically used.

As a rule-based computational process model, ALBATROSS (A Learning-based Transportation Oriented Simulation System) is a fully operational activity-based model developed in the Netherlands. It employs a sequential decision process to generate daily activity schedules of individuals. The sequential decision process uses 26 decision steps, where at individual decision step a CHi-squared Automatic Interaction Detector (CHAID) based induction tree method is utilized. However, a decision process containing 26 decision trees, where each decision tree contains many condition variables is a complex process.

Investigations of the complexity of the decision process model in ALBATROSS have been undertaken. For example Moons, et al (2005) conducted a study to investigate complex and simple classifiers within ALBATROSS. Simple models

include One Rule (OneR) and Feature selection techniques. On the other hand, the applied complex models were the original CHAID and C4.5 decision trees and Support Vector Machines (SVM). The study concluded that simple classifiers do not outperform complex models but are not inferior to complex models. In addition, the study revealed that different decision tree induction methods such as (CHAID, C4.5, CART etc.) achieved comparable results. Janssens et al., (2004) found that Bayesian networks performed better than CHAID decision trees in ALBATROSS. It was revealed that Bayesian networks are better suited to capture the complexity of the decision process model, since they take into account the interdependencies among the variables and decision steps outcome. Keuleers et al., (2001) did experiments in ALBATROSS by using Association Rules as classification rules. However, the research explained above was conducted on an earlier version of ALBATOSS, where the scheduling process model contained only 9 decision steps. And the process model does not contain *activation dependencies*, which encompasses different execution routes in the process model. While the current version of ALBATROSS employs 26 decision steps that are necessary to predict activity schedules for each person under study. Moreover, in the current version of ALBATROSS, the scheduling process model contains two interesting features:

1. Activation dependency, in which the execution of the process model can take many paths depending on the outcome of decision steps.
2. Attributes interdependencies, where the outcome of decision steps are included in the attribute set of subsequent decision steps.

Hence, the complexity of the process model has increased. Therefore, it becomes essential to investigate the complexity of the current scheduling process model implementation. In addition, the sequence and order of the scheduler in ALBATROSS needs to be investigated for the purpose of reducing the complexity of the model.

The analysis in this work is developed within the FEATHERS (Forecasting Evolutionary Activity-Travel of Households and their Environmental RepercussionS) framework. The FEATHERS framework is developed to facilitate the development of modular activity-based models for transportation demand (Bellemans et al., 2010). In this study, Flanders (Belgium) is used as the study area. The scheduling engine that is currently implemented in the FEATHERS framework is based on the scheduling model that is present in the ALBATROSS system.

1.2 Main Research and development Contribution of the Dissertation

ALBATROSS is one of several operational micro-simulation models of travel demand. It aims to predict activity schedules using a sequential scheduling process executed using 26 decision steps. This dissertation discusses three particular contributions with respect to the ALBATROSS model. The first contribution is related to improving the predictive performance of the scheduling process models integrated in FEATHERS / ALBATROSS (FEATALB). Improving the predictive performance can be achieved by considering three major factors.
The first factor is to ensure that the data are of good quality i.e. obtaining better data that are used to train the models at individual decision steps. The second factor involves utilizing better classifiers at individual decision steps that constitute the scheduling process model, which will be referred to as the process model throughout the thesis. The third factor is to attain a better data representation in the context of reordering the preference of decision steps in the process model.

The data requirements for activity-based models are in general demanding compared to conventional travel demand models. Especially, that this type of micro-simulation models should be able to predict the travel behaviour in detail including how the activities are selected and scheduled. Moreover, several studies were conducted to experiment simple and complex classifiers to serve

the purpose of improving the predictive performance. Nevertheless, to date, studies on investigating the data representation in the scheduling process model do not exist.

The current sequence process model in ALBATROSS is based on expert's opinion. Therefore, three data representation (modeling the decision steps simultaneously, the fully-informed and non-informed representations) are investigated to assess the predictive performance of the work activity process model in FEATALB.

The second contribution involves identifying performance boundaries for rule-based activity forecasting models. This will be achieved by obtaining a simple yet sensible model that will serve as the lower performance bound (base line). Furthermore, the research identifies an upper performance bound by optimizing more complex and well-known classification methods that were already described in the literature.

The third contribution is to conduct a sensitivity analysis of the models at each decision step (decision tree models). The sensitivity analysis is performed investigating two important factors used in the decision tree models in FEATALB. The first sensitivity factor involves identifying the ideal number of minimum cases per leaf node while training the decision tree models. Decision tree learning involves setting parameters that are essential in influencing the resulting model's performance. An important parameter is the minimum number of training instances at leaf nodes. Increasing the minimum number of instances at leaf nodes avoid the occurrence of model over-fitting. In ALBATROSS this number was set to 30, however, this number was set using the Dutch data. For the Flemish data, a different number might improve the performance. The second sensitivity factor is the action assignment rule used in predicting values at decision steps. ALBATROSS suggests a probabilistic action assignment rule which considers the probability of predicting a specific class. Rather than

predicting a class variable according to the plurality rule. To read more on the action assignment rule in ALBATROSS, refer to Chapter 3.

The FEATHERS framework is developed as a modular activity-based model for transportation demand in Flanders (Belgium). The process model that is currently implemented in the FEATHERS framework is based on the process model that is present in the ALBATROSS model. The FEATALB framework is fully operational at the level of Flanders, and the models in the process model are based on CHAID decision trees. Furthermore, the FEATALB framework is developed such that additional classifiers can be used. This flexibility allows conducting experiments with different types of classifiers.

1.3 Organization of the thesis and subsequent chapters

The thesis is structured in two main parts. Part one explains setting the scene and development of the experimental laboratory. Also the techniques used in the experiments are explained. In chapter 2 the FEATHERS framework and ALBATROSS model are discussed in details. We elaborate on the extension of the framework to integrate additional classification models. Implementation using a data mining modeling standard is further demonstrated. In chapter 3, classification methods used to train the models and deployed for simulation are explained. Moreover, the action assignment rules and the derivation of rules from induction methods, and feature selection techniques are described. Chapter 4 then gives a detailed description of the data sets and attributes that are used for training the models. Additionally, provides basic statistics on the distribution of the variables and the observed activity patterns and Origin-Destination trips matrices.

The second part of the thesis discusses the research experiments and results. Chapter 5 represents the research design and the methodologies in which the aims of the dissertation are achieved. Next, in chapter 6, the work activity process model is experimented using different classifiers for the purpose of

identifying performance bounds is performed. Followed by chapter 7 where the introduction of alternative process models and their performance is illustrated. Then in chapter 8 a sensitivity analysis of decision tree models utilized at individual decision steps is performed. Finally the thesis concludes with chapter 9 with the final discussion, final conclusions and future research.

References

Timmermans, H.J.P., Arentze, T.A. and Joh, C-H. (2002). Analyzing space-time behavior: new approaches to old problems. Progress in Human Geography, 26, 175-190.

Ettema, D., and Timmermans, H. (1997). Theories and models of activity patterns. In: Activitybased approaches to travel analysis, eds. D. Ettema and H. Timmermans, 1-36. Pergamon, Oxford.

Jones, P.M., M.C. Dix, M.I. Clarke and I.G. Heggie, (1983). Understanding travel behaviour. Gower, Aldershot.

Recker, W.W., M.G. McNally and G.S. Root, (1986). A model of complex travel behavior: Part I - Theoretical development. Transportation Research 20A (4): 307-318.

Recker, W.W., M.G. McNally and G.S. Root, (1986). A model of complex travel behavior: Part II - An operational model. Transportation Research 20A (4): 319-330.

Janssens, D., Wets, G., Brijs, T., Vanhoof, K., Timmermans, H.J.P., and Arentze, T.A. (2004). Improving the performance of a multi-agent rule-based model for activity pattern decisions using Bayesian networks. Transportation Research Record, Vol. 1894, pp. 75-83.

Pendyala, R.M., Kitamura, R. and Reddy, D.V.G.P. (1995). A rule-based activity-travel scheduling algorithm integrating neural networks of behavioral adaptation. Paper presented at the EIRASS Conference on Activity-Based Approaches, Eindhoven, The Netherlands.

Pendyala, R.M., R. Kitamura and D.V.G.P. Reddy (1998). Application of an activity-based travel demand model incorporating a rule-based algorithm. Environment and Planning B, 25, 753-772.

Pendyala, R.M. (2004). FAMOS: Application in Florida. Paper presented at the 83rd Annual Meeting of the Transportation Research Board, Washington, D.C., USA.

Arentze, T.A., and Timmermans, H.J.P. (2000). ALBATROSS: A Learning-based Transportation Oriented Simulation System. EIRASS, Eindhoven University of Technology, The Netherlands.

Arentze, T.A. and H.J.P. Timmermans (2005). Albatross 2: A Learning-Based Transportation Oriented Simulation System, European Institute of Retailing and Services Studies. Eindhoven, The Netherlands.

Moons, E. (2005). Modeling Activity-Diary Data: Complexity or Parsimony? PhD dissertation. Limburg University, Diepenbeek, Belgium.

Janssens, D., Wets, G., Brijs, T., Vanhoof, K., Timmermans, H.J.P., and Arentze, T.A. (2004). Improving the performance of a multi-agent rule-based model for activity pattern decisions using Bayesian networks. Transportation Research Record, Vol. 1894, pp. 75-83.

Keuleers, B., G. Wets, T. Arentze, and H. Timmermans (2001). Association Rules in Identification of Spatial -Temporal Patterns in Multiday Activity Diary Data. In Transportation Research Record: Journal of the Transportation Research Board, No. 1752, TRB, National Research Council, Washington, D.C., pp. 32–37.

Moons, E. A. L. M. G, Wets, G., Aerts, M., Arentze, T.A. and Timmermans, H.J.P (2004). The Impact of Simplification in a Sequential Rule-Based Model of Activity Scheduling Behavior, Forthcoming in Environment & Planning A.

Wets, G., Vanhoof, K., Arentze, T. A., Timmermans, H. J. P. (2000). Identifying Decision Structures Underlying Activity Patterns: An Exploration of Data Mining Algorithms Transportation Research Record, 1718, pp. 1-9.

Chapter 2

The FEATHERS/ALBATROSS (FEATALB) Framework / Model

The purpose of this chapter is to introduce the FEAHTERS framework. Firstly, the ALBATROSS model and its basic components are explained. This includes discussing the scheduling process model, which is the core of the system that controls the scheduling processes. In order to increase flexibility we introduce the FEATHERS framework, in this framework we can add new functionalities and execute new research experiments. For example, thanks to the modular design of the FEATHERS framework we can easily use various classification models. In the framework we also developed different tools like model validation criteria to evaluate model's performance and calculating Origin-Destination matrices of different dimensions. The rest of the chapter is arranged as follows. The ALBATROSS model system flexibility requirements are elaborated, which are then considered in the implementation of FEATHERS. Then, the FEATHERS framework is introduced, which is currently based on the scheduling process model that is present in ALBATROSS. Next explanation on how the FEATHERS framework modular design is extended to be able to use different induction methods. Consequently, the development of model validation criterion and implementation issues is discussed. And finally the chapter ends up with the conclusions.

2.1 The ALBATROSS SYSTEM

In the past few decades, many studies have been conducted in order to try to understand the nature of travel demand. Travel demand is derived from the human needs to participate in activities that are distributed in time and space. Models that simulate travel demand using an activity-based approach have been gaining growing attention in recent times due to their strong behavioral foundation and insightful theoretical demand. Recognizing that travel is a

demand derived from the individuals' needs to perform activities, researchers in travel demand modeling have become increasingly interested in analyzing and predicting individuals' decisions about activity participation. Activity-scheduling models share the objective to predict the sequence of decisions that leads to an observed activity pattern of households/individuals. Activity-based models aim at predicting on a daily basis and for individuals which activities are conducted, by whom, for how long, at what time, the location, and which transport mode is used when traveling is involved (Arentze and Timmermans, 2000). Most activity-based modeling systems are either based on a system of econometric models, or are rule-based models based on a system of rules and heuristics (e.g., ALBATROSS). Some of the tools applied in rules-based models include decision trees, neural networks, informal map analysis and trend surface analysis.

ALBATROSS is a fully operational activity-based model incorporating household decision making (Arentze and Timmermans, 2000, 2004, 2005). It is a rule-based computational process model developed for The Dutch Ministry of Transportation, Public Works and Water Management. ALBATROSS differs from other models, which use utility maximization as a framework for modeling activity scheduling decisions. In contrast, ALBATROSS uses IF-THEN rules as a formalism to represent and predict activity-travel choices of individuals and households. The decision rules are extracted from activity diary data in the form of a decision tree by using a CHAID-based decision tree induction method. In ALBATROSS a sequential decision process model is assumed to generate a schedule.

To generate a schedule for each person, for each day, a sequential decision process is assumed. Decision rules are derived from 26 decision trees, and the activity scheduling process model consists of four components or sub models (Anggraini et al. 2007). The first component is responsible for generating primary work activities and their start time, duration of each work episode if more than one episode is predicted, their location, and the transport mode for the work trip.

The second component is used to generate secondary fixed activities, typically work-related, such as bring/get, business or other mandatory activities. In addition, this decides which type of activities is performed, the number of episodes for each activity, and their start time and duration. The third component is similar to the second component, except that the former determines the flexible activities in the schedule. Flexible activities are those that may or may not be included in the schedule. The fourth and last component is in charge of predicting the transport mode of secondary fixed and flexible activities. It is important to note that in ALBATROSS, the activity travel behavior of the two heads of the household only is captured. These main components assume a sequential decision process in which key choices are made and predefined rules delineate choice sets and implement the choices made in the current schedule. Interactions between individuals within households are to some extent taken into account by developing the scheduling processes simultaneously and alternating decisions between the persons involved.

2.1.1 The ALBATROSS Scheduler Process Model

The ALBATROSS system applies a fixed sequence of decision models, which forms the ALBATROSS scheduling process model (Arentze and Timmermans 2004). In the later part of this thesis when we refer to the process model we mean the scheduling process model. The applied decision models in ALBATROSS are rule-based models. These rules are accommodated in the rule-based scheduling engine to infer individuals' activity schedules at the household level. In addition, these rules take into account different space and time aspects, possible scheduling constraints, as well as decision trees derived from individuals' daily activity-travel diaries. The schedules are generated in the scheduling engine, a fixed sequential decision process is assumed in which fixed activities such as work and other fixed activities are scheduled prior to flexible activities. Furthermore, details about each activity, its starting time, duration, trip-chaining, location and transport mode choice (if needed) are derived in a priority-based sequential order. Fixed activity related to work is considered in this thesis. To schedule the activities, household interactions

between individuals as well as constraints are taken into account in the ALBATROSS system. The constraints can be of different types: (1) situational constraints (persons cannot be in different locations at the same time), (2) institutional constraints (opening hours), (3) household constraints (bringing children to/from school), (4) spatial constraints (e.g. particular activities cannot be performed at particular locations) and (5) time constraints (each activity requires a minimum amount of time).

The analyses reported in this thesis are performed on the first component of ALBATROSS, excluding the transport mode decision step. Figure 2.1 depicts the work activity decision process model used in ALBATROSS. Each numbered rectangle refers to a decision tree model derived from activity diary data. In the work activity decision process model only two work episodes can be predicted. A work episode is a periods of time in which the person under study performs a work activity continuously without a break time. The index *j* refers to the number of work episodes, if more than one work activity episode is predicted.

The first decision step evaluates whether the individual's schedule contains a work activity or not. If so, the duration of the work activity can be predicted. Next, the number of work activity episodes is predicted. If two work episodes are predicted, then the ratio between work episodes and the break time duration decision steps are executed. Finally, the work activity start time is predicted. Decision steps 1 and 3 are discrete choices, whereas, decision steps 2, 4, 5 and 6 are continuous choices. It is noteworthy that if decision step 1 infers no work episode, then decision steps 2-6 will not be executed. Similarly, if decision step 3 evaluates to one work episode, then decision steps 4 and 5 will not be evaluated. This means that there is an activation dependency in the execution of this process model.

The rules at decision steps in the scheduler decision process model in ALBALTROSS are derived using a Chi-squared Automatic Interaction Detector (CHAID)-based induction method. CHAID is applied to generate decision trees trained from activity-travel diary data. This means that the chi-square measure is

used to successively split condition variables to find homogeneous sets until a stop criterion is met. This process is represented in a decision tree. Decision tree concepts and other induction methods used in this thesis are discussed in details in Chapter 3.

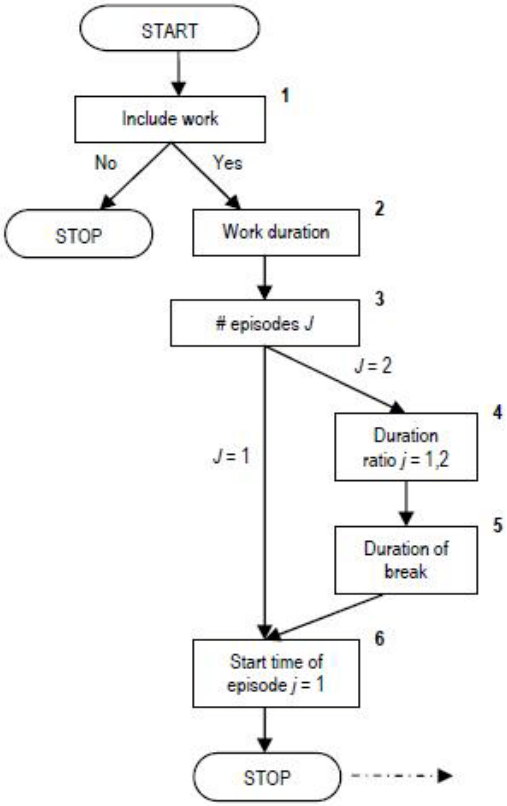


Figure 2.1 Work activity process model in ALBATROSS, adapted from Arentze and Timmermans (2004).

In the next section, the FEATHERS framework emphasizing on its modular design is discussed. Subsequently, the expansion of the framework to adopt various induction methods is explained. Then the data used to train the models is discussed in details. Finally, the derivation of decisions from induction methods is explained.

2.2 The FEATHERS Framework

The FEATHERS (Forecasting Evolutionary Activity-Travel of Households and their Environmental RepercussionS) framework is developed to facilitate the development of a modular activity-based model for transportation demand in Flanders (Belgium). At first the framework adopted a four-stage development trajectory, for a smooth transition from the four-step models towards static activity-based models in the short term and dynamic activity-based models in the longer (Bellemans et al. 2010). In this study, Flanders (Belgium) is used as the study area.

In order to include ALBATROSS in FEATHERS, the model parameters must be modified to fit the Flemish data and situation. The modification of the model parameters and the decision models, involved gathering data for the Flemish region and training and replacing each of the 26 decision trees. To achieve this, the ALBATROSS components and model parameters have been studied in details and later adapted to fit the Flemish situation, for more on this refer to Kochan, 2012. However, some practical limitations were determined that restrained further experimentations and there was a need for new implementations. The following two sections, describe this development and implementation work in more details.

The original ALBATROSS model is developed using Borland C++. This prevented further implementation and impedes adding new functionalities. Hence, while reverse engineering the model, a porting process to Microsoft Visual C++ is performed.

Some parts of the model were re-implemented. The implementation involved using technologies to boost the design of experiments conducted in this thesis. Technologies, such as incorporating a database with the system to capture the output (predicted) schedules of persons. Using a database enhances conducting experiments by being able to generate statistics of the predicted schedule without having to run the simulation again.

Next we added new functionalities. First, in ALBATROSS, at each decision step in the process model is controlled by a CHAID decision tree. The process model contains 26 decision trees which are hard coded into the system. This approach makes experimenting with other induction or classification methods inapplicable. Therefore, the platform is extended to employ other induction methods.

Second, to generate outcome statistics and Origin-Destination (OD) matrices, the simulation must be run. So in case extra statistics or ODs with extra dimensions need to be generated, the simulation must be run, which is a time consuming process. To overcome this limitation the outcome (predicted) schedule information is captured and stored in a database system. Which allows for generating extra statistics and ODs by introducing a Statistical Module (StatMod) that is configurable to generate several statistics and ODs with many dimensions from a the database version of the schedule without needing to run the simulation.

Third, the validation of the models at the individual classifier, the activity pattern, and the spatial and temporal levels are also implemented. The models at the individual classifier level are validated using Confusion Matrix statistics and the Brier Score. Where at the activity pattern level, the models are validated using the Sequence Alignment Method (SAM). And at the spatial temporal level, OD matrices and other statistics are calculated. Model validation is discussed in details in chapter 3.

Figure 2.2 represents an overview of the FEATHERS framework and its modular design. The interactions between modules are depicted by arrow lines. As can be seen in the Figure, the ALBATROSS model is integrated as part of the Schedule module (SchedMod). SchedMod is a generic module where other models can be integrated. In the following subsections important modules used within this study are discussed.

Configuration module (ConfMod)

To take full advantage of the modular design of FEATHERS, flexible configuration functionality is necessary. ConfMod maintains a configuration file,

in which each module is included with its own setting. As shown in Figure 2.2 all modules in the system communicates with ConfMod in order to employ a specific setting while running the system. The configuration file stores all the configuration settings for the FEATHERS modules in XML format (W3C, 2006). Using XML allows for ease of adding or modifying new modules and/or module setting(s), since it is human readable. In addition, XML is supported by most programming languages. Each module can be switched on (active) or off (inactive). And this allows users to perform experiments without having to load unnecessary modules or functionalities.

Each module in the configuration file contains its own settings and parameters that necessary to be provided while running the simulation. Hence, ConfMod was implemented to easily browse through the different levels up and down, using a single instance of ConfMod to ensure file access consistency.

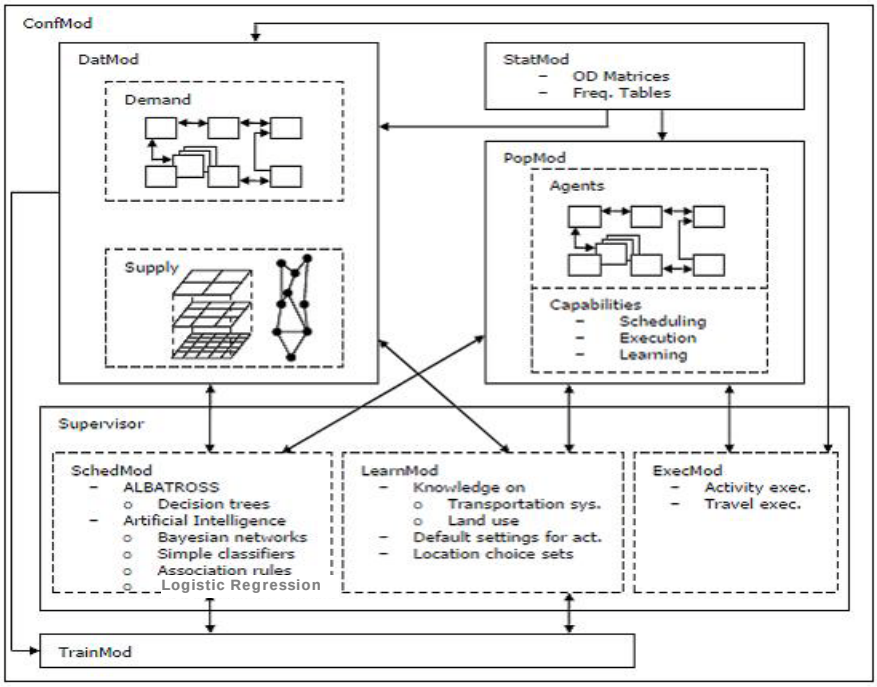


Figure 2.2 A schematic overview of the FEATHERS modules, their functionalities and interactions, adapted from Bellemans et. al. (2010)

Data module (DatMod)

This is a core module in the FEATHERS framework. It provides access to the data that will be accessible by all other modules. As shown in Figure 2.2, DatMod contains two main data types: supply and demand data. The supply data includes the transportation network and information on geographical zones in the area under study. On the other hand, the demand data consist of the activity-travel diaries or schedules that describe the demand for the execution of activities at certain locations as well as the resulting demand for transportation. Both the supply and the demand data managed by the data module are made available to other modules through the data module's standardized interface.

Population module (PopMod)

In order to perform a simulation of activity and travel behavior of individuals in a population, a synthetic population consisting of persons and households (and optionally cars belonging to the household) needs to be built. The population module is responsible for the management of the different agents (persons) that are used in the synthetic population. The synthetic population therefore consists of a collection of agents where each agent is characterized by a number of attributes. As mentioned previously, the data required are available at population level in Flanders by means of the socio-economic survey.

Statistics (StatMod)

The statistics module provides reports regarding the (synthetic) population and the activity-travel schedules to the FEATHERS user. This includes information that can be extracted on the households' level (e.g. distribution of households according to availability of means of transportation); persons (e.g. usage of transportation modes), journeys (e.g. average number of journeys per day); lags (e.g. average number of lags per journey) and activities. Given the similarity in the person, household, car, activity, journey and lag entities and their relations in both the data module and the population module, the statistical module and the visualization module make abstraction from the fact whether they consult the

data module or the population module to extract the data to report to the user. Hence, statistics that are implemented for the survey data in the data module can readily be used to draw the corresponding statistics on simulated data from the population module. The statistics are to be drawn by the statistical module is configured through the configuration module. As the activity-travel diaries contain detailed travel information, the statistical module provides the functionality of scanning through all schedules and compiling an OD matrix. With the functionality of storing predicted schedules in a database, StatMod can generate schedule statistics without having to run the simulation. Given the level of detail of the data, the travel information can be aggregated in segmented OD matrices such as time sliced OD matrices, OD matrices per transportation mode, and OD matrices per activity type. In addition, the spatial level of detail can be selected while calculating OD matrices. In FEATHERS three layers are provided, subzone, zone and superzone levels which corresponds to the spatial level of which the travel information is provided. The OD dimensions and levels are discussed in more details in section 2.6.

Schedule module (SchedMod)

The schedule module is a generic module in which different process models and different decision models can be implemented. Determining which process model is used for simulation is activated in the configuration module. The schedule module is tightly interfaced with the population module as it implements the process model that uses input data from the population module and stores the results in the schedules in the population module.

The process model that is currently implemented in the FEATHERS framework is based on the process model that is present in the ALBATROSS system, which will be referred to as the FEATALB framework. This means the 26 CHAID based decision trees are implemented in FEATHERS as one process model implementation. Each decision tree is used to model decisions on specific activity (e.g. going to work) and its properties such as, duration, start time, and transport mode for work journey. In addition SchedMod contains algorithms to make

schedules consistent, taking into consideration all types of constraints. Consequently, considering the analyses performed in the context of this work, the FEATHERS framework is extended to conduct experiments using alternate induction methods, such as decision tree, logistic regression and OneR (Holte 1993) on one hand. And to be able to change the order of decision steps in the work activity process model on the other hand. Therefore the DecisionMaker class is integrated in the SchedMod to facilitate employing other data mining methods within FEATALB.

In the next section, the extension of the FEATALB framework and its implementation to accommodate additional induction or classification methods is discussed in details. Additional induction methods include decision trees (discrete and continuous) other than CHAID, Logistic regression and OneR. The development of calculating Origin-Destination matrices at different zoning levels and dimensions is further explicated. Then the implementation of SAM within FEATHERS is explained.

2.3 Open architecture of the FEATALB Framework to use different Induction methods (PMML functionality)

To overcome the limitation of using induction methods other than CHAID, it was required to extend the FEATALB framework. Furthermore, to be able to experiment the model with other induction methods, a portable mechanism is required to be integrated within FEATALB. This mechanism allows users to have the functionality to choose a specific induction method at any decision step in the ALBATROSS in the chosen process model. The additional functionality allows us to train models outside FEATALB, using data mining packages that can export Predictive Model Markup Language (PMML) (Guazelli et al. 2010). PMML is an XML based language to annotate data mining model parameters in textual form with meta-data for re-use. It is developed by the Data Mining Group (DMG) to provide a way for applications to describe models related to predictive analytics and data mining and to share those models between PMML-compliant applications (http://www.dmg.org/v4-0). Thus, using this functionality, the CHAID

induction method is replaced by other decision tree methods or alternative induction methods. Namely C4.5 decision tree (Quinlan 1986), or any other decision tree based induction within the scheduling model. The modular design of FEATHERS allows for easy integration, thus, the DecisionMaker class is implemented as an abstract class for other induction methods classes. This class is used at any place in the code to evaluate specific decision step using any of the implemented induction method in PMML format. In addition, a polymorphic function is facilitated in the implementation to provide flexibility while invoking a specific induction method at any decision step.

Figure 2.3 shows the class diagram of the DecisionMaker class. The DecisionTree, Regression and OneR classes inherits functionalities from the DecisionMaker class. Using the DecisionTree class implementation, any decision tree model in PMML format can be utilized and used. The LinearRegression class can employ Linear and Logistic regression PMML models. And similarly, the OneR class employs One Rule induction methods.

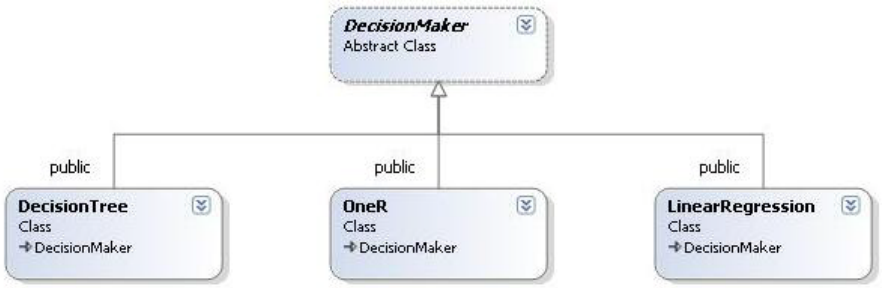


Figure 0.3 DecisionMaker class diagram

Another issue related to extending FEATHERS to use different induction methods is adding relevant sections in the configuration file. So, extra information related to available induction methods and paths to models for each decision step is added accordingly.

2.3.1 Predictive Model Markup Language (PMML)

PMML is an independent method of defining models so that incompatibilities are no longer a burden to the exchange of models between applications. It allows users to develop models within any third party application and use other applications to visualize, analyze, evaluate or deploy the models. Previously, this was very difficult, but with PMML, the exchange of models between compliant applications is straightforward. PMML is an XML-based standard, so the specifications come in the form of an XML-schema. As shown in Figure 2.4, PMML is composed of several elements which summed different functionality as it relates to the input data, model, and outputs (Guazelli et al. 2009).

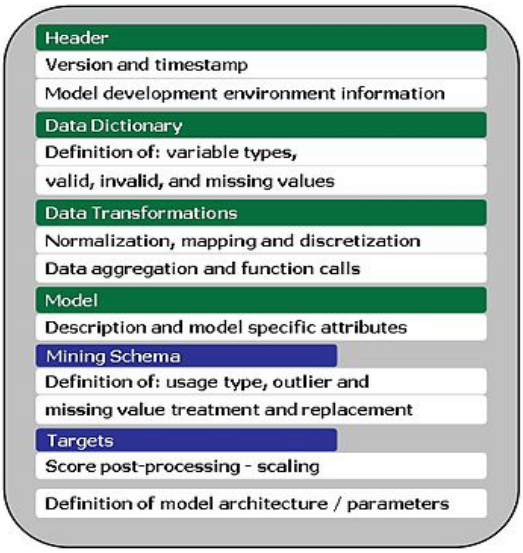


Figure 2.4 PMML overall structure described sequentially from top to bottom. Adapted from Guazelli et al. (2009)

The header element contain general information about the PMML document, such as copyright information for the model, its description, and information about the application used to generate the model such as name and version. The data dictionary element contains definitions of variable types, valid, invalid, and missing values. Variable types are defined as, continuous, categorical, or ordinal. The Data transformation element is responsible for mapping user data into other

desirable forms that can be used by the data mining method. Several data transformation are defined in PMML: normalization (e.g. Maps values to numbers), Discretization (e.g. mapping continuous values to discrete values), functions (e.g. derive a value by applying a function to one or more parameters), and aggregation (e.g. Summarizes or collects groups of values).

In the Model element, the definition of the data mining model is specified. In addition the model name, function name (classification or regression) and technique-specific attributes are defined. Followed by the model representation, this begins with the Mining Schema, which lists all fields used in the model. The Model element also contains the Targets element, which allows for the scaling of predicted variables.

2.4 Model Validation

Model validation is an important step in the model building sequence. The ALBATROSS model is validated at three levels. First, the individual classifier step using confusion matrix accuracy statistics. Secondly, the activity pattern levels using SAM to calculate how close predicted to the observed activity pattern sequence are. Thirdly, at the spatial level, by calculating correlation coefficient between observed and predicted OD matrices.

As a new functionality, the validation of FEATHERS framework is performed outside the system. To validate the newly integrated induction methods, effectively and automatically, it was necessary to implement them inside the DecisionMaker architecture. Similarly, calculating the SAM distance to validate the model at the activity pattern level is developed by adding the SAM class implementation. The SAM class implementation required extracting activity pattern symbols from observed and predicted schedules. The output SAM distances are reported in a text file, by calculating the average distances for all cases under study. Finally, to gain more understanding of the model, and be able to conduct more experiments, ODs for several dimensions at all spatial layers are

developed. The development of the OD generation is included in StatMod. The travel information is aggregated in segmented OD matrices, ODs per each hour of the day, per transport mode, per activity type. Additionally, using the configuration file functionality, ODs of more detailed dimensions can be generated i.e. ODs for each hour of the day per transport mode, per gender, per activity type, per age group etc. The implementation of generating ODs is performed from the predicted database setting, so the simulation is run once, and ODs may be generated at anytime later.

2.5 Conclusions

Many activity-travel demand micro-simulation models have become operational. Moving the currently operational activity-based models into practice is an increased concern (Bellemans et. al, 2010). Some of the operational activity-based models introduced these interdependencies, however, operational micro-simulation of activity-travel demand models have remained limited (Davidson et. al, 2007). There are several reasons for this slow dissemination that can be thought of in this regard. One of the main challenges faced by the travel demand forecasting industry is the ability to quickly plug-in several new theoretical advances in a time and cost effective manner. To explore these theoretical advancements, scientific laboratory experiments are needed, for this reason it is important to depend on a basic platform where these advancements can serve as system add-ons.

Taking the above into consideration, the FEATHERS platform is developed as a modular activity-based travel demand model, where the emphasis is on the practical use of the system by practitioners and end users. The platform is modular by design using the object-oriented programming paradigm, which allows for more flexibility. Moreover, this framework also allows for rapid employment of activity-based models for new study areas so that the threshold for these kind of models shrinks tremendously. In line with the latter advantage, with this general simulation framework, any activity-based travel survey can be

used with a minimum of processing time in order to train/re-train the transport model inside the framework.

The ALBATROSS model system and its main components were explained. The ALBATROSS system limitations from an implementation point view were then discussed. Three limitations are found, (1) the ability to employ induction methods other than the CHAID decision trees at each decision step. (2) Generating statistics without having to run the simulation, and (3) Validating the model while running the simulation for model comparison and experimentation purposes.

To overcome these limitations The FEATALB framework was developed to facilitate the development of modular activity-based models for transportation demand. The FEATALB framework was extended to facilitate the use of various induction methods using PMML, a data mining modeling standard. This allows studying the effect of introducing new sequential process models and decision models, also allows for replacing the original CHAID-based decision trees by other induction methods.

References

Anggraini R., Arentz T., and Timmermans H. Modeling car allocation decisions in automobile deficient households, in: Proceedings of he European transport conference, Noordwijkerhout, The Netherlands 2007.

Arentze, T.A., and Timmermans, H.J.P. ALBATROSS: A Learning-based Transportation Oriented Simulation System. EIRASS, Eindhoven University of Technology, The Netherlands, 2000.

Arentze, T.A., and Timmermans, H.J.P. Measuring Impacts of Condition Variables in Rule-Based Models of Space-Time Choice Behavior: Method and Empirical Illustration, Journal of Geographical Analysis, 2003a, 35, 24-45.

Arentze, T.A., and Timmermans, H.J.P. Measuring the goodness-of-fit of decision-tree models of discrete and continuous activity-travel choice: methods and empirical illustration, Journal of Geographical Analysis, 2003 b, volume 5, Number 2, 185-206.

Arentze, T.A., and Timmermans, H.J.P. A Learning-Based Transportation Oriented Simulation System. Transportation Research Part B, 2004, 38, 613-633.

Arentze, T.A., and Timmermans, H.J.P. ALBATROSS 2: A Learning-Based Transportation Oriented Simulation System. European Institute of Retailing and Services Studies. Eindhoven. The Netherlands, 2005

.

Bellemans T., Janssens D.., Wets G., Arentze, T., and Timmermans H. Implementation Framework and Development Trajectory of the FEATHERS Activity-Based Simulation Platform. Proceedings of the Annual Meeting of the Transportation Research Board, 2010.

Data Mining Group website | PMML 4.0 (http://www.dmg.org/v4-0)

Davidson, W., Donnelly, R., Vovsha, P., Freedman, J., Ruegg, S., Hicks, J., Castiglione, J., and R. Picado. Synthesis of first practices and operational research approaches in activity-based travel demand modelling. Transportation Research Part A, No. 41, Transportation Research Board of the National Academies, 2007, pp. 464-488.

Guazelli, Alex, Michael Zeller Wen-Ching Lin and Graham Williams (2009). PMML: An Open Standard for Sharing Models, The R Journal Vol. 1/1, May 2009, pp. 60-65.

Guazelli, Alex, Wen-Ching Lin and Tridivesh Jena. (2010) PMML in Action: Unleashing the Power of Open Standards for Data Mining and Predictive Analytics. CreateSpace. 2010, ISBN 978-1452858265, pp 5.

Holte, R. C. Very simple classification rules perform well on most commonly used datasets. Machine Learning, 1993, 11(1):63–90.

Janssens, D., G. Wets, T. Brijs, K. Vanhoof, T. A. Arentze, and H. J. P. Timmermans Integrating Bayesian Networks and Decision Trees in a Sequential Rule-Based Transportation Model. European Journal of Operational Research, 2006, Vol. 175, No. 1, pp. 16-34.

Joh, C-H, Arentze, T.A., Hofman, F. and Timmermans, H.J.P. Activity-travel pattern similarity: a multidimensional alignment method. Transportation Research B, 2002, 36, 385-403.

Kochan, B. (2012) Implementation, validation and application of an activity-based transportation model for Flanders, PhD dissertation. Hasselt University, Diepenbeek, Belgium, 2012.

Moons, E. (2005) Modelling Activity-Diary Data: Complexity or Parsimony? PhD dissertation. Limburg University, Diepenbeek, Belgium, 2005.

Quinlan, J .R.. Decision trees and multi-valued attributes. In J .E. Hayes & D. Michie (Eds.), Machine intelligence 11. Oxford University Press (in press), 1985.

Quinlan, J.R. Induction of Decision Trees, Machine Learning, 1986, vol. 11, no. 1, pp. 81–106.

Chapter 3

Essential Components, Classifiers, Derivation of decisions from classifiers, Model validation and model comparison criteria

The aim of this chapter is to introduce the essential components of the analyses, classifiers, experimentation and models validation and comparison criteria used in this thesis. A survey of simple and complex classification methods used in FEATALB as a rule-based activity based model are discussed. The difference between discrete choice and continuous classification methods is further explored. Then the classification methods used in this dissertation, namely, Decision trees (CHAID, C4.5, and CART), One Rule (OneR), Logistic Regression, and Multi-Target Info Fuzzy Networks (M-IFN). For each classification method, the learning process is based on a specific statistical approach. Therefore, the action assignment rule, which is the selection of a value of the response variable to generate a prediction for a given case (referred to as the derivation of decisions from classifiers), is discussed. The derivation of decisions from discrete choice and continuous models are elaborated separately. Models comparison criteria and validation levels are then explained, as the different classifiers are used in each decision step in the scheduler process model. The validation levels are introduced to demonstrate models performance taking in account attribute interdependencies between decision steps. Finally, for purposes serving the analyses introduced in this thesis, the attribute selection and discretization methods are discussed.

3.1 Introduction

In the past few decades, research has been conducted in order to try to understand the nature of travel demand. Travel demand is derived from the human needs to participate in activities that are dispersed in time and space. Recognizing that travel is a demand derived from individuals' needs to perform

activities, researchers in travel demand modeling have become increasingly interested in analyzing and predicting individuals' decisions about activity participation. Activity-scheduling models share the objective to predict the sequence of decisions that leads to an observed activity pattern of households/individuals. Activity-based models aim at predicting on a daily basis and for individuals which activities are conducted, by whom, for how long, at what time, the location, and which transport mode is used when traveling is involved (Arentze and Timmermans, 2005). The utility-maximization modeling assumes that individuals make their activity-travel decisions to maximize the utility derived from the choices they make (Timmermans et. al, 2002). However, this approach has been argued by scholars that individuals do not necessarily arrive at 'optimal' choices (Arentze et. al, 2001). Conversely individuals use context-dependent heuristics (Timmermans et. al, 2002). Alternatively, computational process activity-based models formalize choices of outcomes to such heuristics as rules to predict activity-travel patterns. Rule-based activity-based models have proved to be more flexible than utility-maximising models (Arentze et. al, 2001) and they also perform well in predicting transport choice behaviour if an induction technique is used (Wets et. al, 2000). However, computational process activity-based models are argued to lack ease of interpretation, and hard to statistically assess the decision-rules performance (Moons, 2005). As a result, even though computational process rule-based activity-based models are developed to better reflect the behavioural characteristic underlying activity-travel decisions, such models are viewed as black boxes. Examples of such models include Scheduler (GÄarling et. al, 1989), AMOS (Pendyala et. al, 1995), and ALBATROSS (Arentze and Timmermans, 2000).

As a rule-based computational process model, ALBATROSS is a fully operational activity-based model. It employs a sequential decision process to generate daily activity schedules of individuals in the context of a household. The sequential decision process contains 26 decision steps, where at each decision step CHi-squared Automatic Interaction Detector (CHAID) based induction tree methods are utilized. However, a decision process containing 26 decision trees, where

each decision tree contains many condition variables, and when those condition variables are included in the decision tree rules yields a complex process model.

Initiatives to investigate the complexity of the ALBATROSS decision process model have been undertaken. A study was conducted to investigate complex and simple classifiers within ALBATROSS by Moons, et al (2005). Simple models include OneR (Holte, 1993) and Feature selection techniques. OneR is a very simple classifier that provides a rule that is based on the value of a single attribute. Further, feature selection techniques aims at reducing the number of irrelevant attributes, which as a consequence reduce the size of decision tree rules. On the other hand, complex models applied were C4.5 decision trees (Quinlan, 1986) and Support Vector Machines (SVM). The study showed that simple classifiers do not outperform complex models but are not inferior to more complex models. Furthermore, models obtained by very simple models may not be applicable in the sense that they do not generate realistic schedules, as obtained by Sammour et al. (2012). Therefore, obtaining a simple model that is applicable is needed. Janssens et. al. (2006) found that Bayesian networks performed better than CHAID decision trees in ALBATROSS. And that Bayesian networks are better suited to capture the complexity of the decision process model, since they take into account the interdependencies among the variables and decision steps outcome in the decision process model. Other classification methods, such as, Association Rules were experimented with ALBATROSS as illustrated by Keuleers et. al (2001). In comparative studies by Wets, et al. (2000) and Moons, et al. (2004) revealed that different decision tree induction algorithms such as (CHAID, C4.5, CART etc.) achieve comparable results.

3.2 Discrete choice and Continuous classifiers

Two main groups of data mining methods are supervised and unsupervised learning methods. Classification is one of the supervised learning methods for data mining that uses predictive approach. Classification is learning a function that maps (classifies) a data item into one of several predefined classes (Weiss

and Kulikowski 1991; Hand 1981). Classifiers are first learned (model construction) and then applied (model usage).

Model construction needs a set of predefined cases where each case belongs to a predefined class. A subset of the cases is used for model construction (training set). And the model is represented as classification rules, decision trees, or mathematical formulae. On the other hand, model usage entails classifying future or unknown cases. Nevertheless, before using a model, it has to be validated by applying the model on an unseen set of cases (test set), where the classes/labels are already known. There are several accuracy measures to signify model validation, for example the accuracy rate, which is the percentage of test set samples that are correctly classified. Validation and model comparison criterion are discussed in details in section 3.5. Some data mining classification algorithms require specific data types and specific content types to be able to function correctly. Data types can be discrete, which is a finite number of values of a tuple with no continuum between values, such as gender. Another type of data is continuous, which means that the column contains values that represent numeric data on a scale that allows intervening values. Unlike a discrete column, which represents finite, countable data, a continuous column represents scalable measurements, and it is possible for the data to contain an infinite number of fractional values. A column of temperatures is an example of a continuous attribute column. Some predictive modeling techniques are more designed for handling continuous predictors, while others are better for handling categorical or discrete variables.

3.2.1 Decision Tree induction methods general concepts

Decision tree models are mainly used in classification because of their ease of construction and usage. The main goal of tree induction is to find a set of rules that best fits the data. Each path from the root of a decision tree to one of its leaves can be transformed into a rule simply by joining the tests along the path, and taking the leaf's class prediction as the class value.

The FEATALB framework uses a probabilistic action-assignment rule, for both discrete and continuous choice decisions, instead of a deterministic assignment rule, because this results in a better prediction of the aggregate distributions (Moons, et. al, 2005). And so, each rule is assigned a probability distribution that is derived from the class frequency distribution. Another important issue in decision tree learning is over-fitting on the data. The concept of over-fitting occurs when the induction algorithm generates a decision tree that perfectly fits the data in the training data set but lacks the capability of generalization of instances not present in the training set. To avoid over-fitting the minimum number of cases at leaf nodes was set to 30 for CHAID, C4.5 and CART decision tree models (Hall, et. al, 2009).

3.2.1.1 CHAID based decision Tree Induction

The CHAID stands for **CH**i-squared **A**utomatic **I**nteraction **D**etector, and was introduced by Kass (1980) as an efficient statistical technique for segmentation. It originated from the automatic interaction detection (AID) method (Morgan and Sonquist 1963).

The CHAID based induction tree method is able to generate trees with more than two branches attached to the same node at any level of the tree and mainly suited for the analysis of large data sets. It is based on the chi-squared (χ^2) statistic to identify the best split of the data set on condition variables into homogenous partitions with respect to the class variable. In addition the CHAID based tree induction method allows for specifying a threshold (α) for splitting of the significance level for the χ^2 value and the minimum number of cases at leaf nodes. The tree building algorithm is performed by recursively iterating through the condition variables to test for each variable the pair of categories whether there is no statistically significant difference within the pair with respect to the class variable. The split with the highest significance value across condition variables is selected. This procedure is repeated until no significant splits are found or the maximum number of cases at leaf node nodes is reached.

The CHAID tre building algorithm proceeds in steps. It first detects the best partition for each predictor. Then the predictors are compared and the best one is chosen. The attributes are subdivided according to this chosen predictor. Each of these subgroups are then re-analysed independently, to produce further subdivisions for analysis. Let the dependent variable Y have $d \leq 2$ response categories, and a particular predictor under analysis X have $c \leq 2$ categories. A subproblem in the analysis is to reduce the given $c \times d$ contingency table to the most significant $j \times d$ table by combining (in an allowable manner) categories of the predictor. The full CHAID algorithm works as follows:

Step 1. For each predictor in turn: cross-tabulate the categories of the predictor with the categories of the dependent variable and do steps 1a and 1b.
Step 1a**.** Find the pair of categories of the predictor (only considering allowable pairs as determined by type of the predictor) whose 2 × d sub-table is least significantly different. If this significance does not reach a critical value, merge the two categories, consider this merger as a single compound category, and repeat this step.
Step 1b**.** For each compound category consisting of three or more of the original categories, find the most significant binary split (constrained by the type of predictor) into which the merger may be resolved. If the significance is beyond a critical value, implement the split and return to step 1a.
Step 2**.** Calculate the significance of each optimally merged predictor, and isolate the most significant one. If this significance is greater than a criterion value, subdivide the data according to the (merged) category of the chosen predictor.
Step 3. For each partition of the data that has not yet been analysed, return to step 1. This step may be modified by excluding from further analysis partitions with a small number of observations (to ensure the validity of the significance test).

For the derivation of the decision models in ALBATROSS the threshold α was set to 0.05 and the minimum number of cases is set to 30 (Arentze and

Timmermans, 2005), therefore, the minimum number of cases per leaf node is also set to 30 for all decision tree induction methods in this research.

3.2.1.2 The C4.5 Decision Tree induction

One popular example of a decision tree construction method is ID3 (Quinlan, 1986) which is based on an information theory approach, in an attempt to minimize the number of tests required to classify a case. An improvement of the ID3 decision tree induction algorithm that is able to handle non-categorical data is the C4.5 algorithm developed by Quinlan, 1993. Like CHAID, C4.5 is not restricted to binary splits and it produces a tree of variable number of decision and/or leaf nodes, and for this reason C4.5 was chosen to be employed in this analysis as it is expected to have more or less the same performance as the CHAID based induction method, which will support the analysis of the work activity process model.

There are two stages for building a classification decision tree in the C4.5 algorithm. The first stage involves generating the decision tree based on the training data set, where the second stage has to do with pruning the decision tree based on the validation or test data set. The algorithm works as follows. Assume we have a date set *S* of training cases or samples, each case consists of *n* condition or explanatory variables $xi_1, xi_2, \ldots xi_n$ and a class or response variable C_i, for $i = \{1,2, \ldots p\}$ classes. C4.5 first grows an initial tree using the divide-and-conquer technique by splitting the training set into homogeneous subsets S1, S2…Sp, until the leaf nodes contain only cases from a single class. An important issue in learning classification trees is over-fitting on the data. The concept of over-fitting is very important in data mining as in decision tree induction. It occurs when the induction algorithm generates a decision tree that perfectly fits the data in the training data set but lacks the capability of generalization of instances not present in the training set. In decision trees over-fitting usually occurs when the tree has too many nodes relative to the training data available. Therefore to avoid over-fitting C4.5 adopts pruning strategy, where the decision tree is simplified by removing one or more subtrees and replacing them with leaves. In the C4.5

WEKA (Witten and Frank 2005) implementation (J48), over-fitting can be avoided also by selecting the minimum number of cases at leaf nodes. This parameter forces the algorithm to stop growing the tree when a preferred number of cases are reached. In the next sections the techniques of splitting or partitioning the data set and the pruning strategy C4.5 employs will be discussed.

Splitting criterion

C4.5 uses two heuristic criteria to split the training cases, the information gain that uses attribute selection measure, which minimizes the total entropy of the subset {Si}, and the default gain ratio that divides information gain by the information provided by the test classes. The information gain criterion is based on information theory. As stated by Quinlan (1993) the information theory on which the gain criterion is based can be explained using the following concepts and definitions:

- Information of a message: The information conveyed by a message depends on its probability and can be measured in bits as minus the logarithm to base 2 of that probability. The information of a message that a random case belongs to a certain

Class C_i is computed as:

$$-log_2\left(\frac{freq(Ci,S)}{|S|}\right)bits \qquad (3.1)$$

Where S is a training set of cases, C_i is a class i, $freq\ (C_i, S)$ is the number of cases in S that belongs to class C_i and $|S|$ is the number of cases in S.

The above definitions forms the basis of the average amount of information needed to identify the class of a case in the training set, which is called the entropy.

- Entropy of a training data set:

$$E(S) = -\sum_{i=1}^{p} \frac{freq(C_i,S)}{|S|} \times log_2\left(\frac{freq(C_i,S)}{|S|}\right) \quad (3.2)$$

Where S a training set of cases, *p* the number of classes, C_i is a class i, *freq* (C_i, *S)* is the number of cases in *S* that belongs to class C_i and |*S*| the number of cases in *S*. Entropy is also measured after that *S* has been partitioned in to *m* sets using the outcome of a test carried out on attribute *X*. This gives:

- The Entropy after the training set has been partitioned on a test *X*:

$$E_X(S) = \sum_{i=1}^{m} \frac{|S_i|}{|S|} \times E(S_i) \quad (3.3)$$

Using these two measures the information gain, which means how much information, can be gained by branching on attribute *X* can be computed as follows:

$$Gain(X) = E(S) - E_X(S) \quad (3.4)$$

The C4.5 algorithm is an enhancement over the ID3 decision tree induction algorithm to handle non-categorical and missing data. In ID3, the split test selected is the one which maximizes information gain because it is expected that the remaining subsets in the branches will be the easiest to partition. However the gain criterion has one drawback, namely the information gain applied to attributes that can take on a large number of distinct values might learn the training set too well. Therefore, in C4.5 an adapted form of information gain is employed. This criterion is called the gain ratio, where in this criterion the gain attributable to tests with many outcomes is adjusted using some kind of normalisation. This indicates the information generated by partitioning *S* into *m* subsets. Consequently, the *split info*(*X*) measurement has to be defined.

$$split\ info(X) = -\sum_{i=1}^{m} \frac{|Si|}{|S|} \times log2\left(\frac{|Si|}{|S|}\right) \quad (3.5)$$

The gain ratio corresponds to how much of the gained information is useful for the classifier; consequently, C4.5 will select the test that has the maximum gain

ratio. After the decision tree is built, pruning takes place, which in turn will simplify the decision tree by eliminating one or more subtrees and replace them by leaves.

Pruning

Pruning is useful for decision trees as it improves generalization and accuracy of unseen test instances. C4.5 uses an approach called pessimistic pruning. In this approach the decision tree is evaluated on the training data set, it was also proposed by Quinlan (1993), and was developed in the context of ID3. Quinlan found that it is too optimistic to use a training set to test the error rate of a decision tree, because decision trees have been customized to the training set. In this case, the error rate can approach 0. But if some data other than the training set is used; the error rate will increase dramatically. To solve this problem, Quinlan used continuity correction for the binomial distribution to get an error rate. If a given branch has a higher error rate than a simple leaf, the branch is replaced with a leaf. This heuristic is applied to the decision tree from bottom to top. The error rate is calculated in the following manner (Quinlan 1993). If *n* training examples are covered by a leaf, where the number of incorrect examples is *e*, the algorithm considers this as a sample in which *e* events are observed from *n* trials. To determine the predicted error rate, C4.5 consequently attempts to predict the probability of an error across the cases covered by a leaf. Since this probability cannot be determined exactly, a posterior probability distribution is calculated using an upper and lower confidence interval.

The C4.5 decision tree model was trained and generated using the Rattle package for R (Williams 2009). The model was then exported to PMML and the *decisionMaker* class is used in the FEATHERS framework to deploy PMML decision trees.

3.2.1.3 Classification and Regression Trees (CART)

CART (Classification And Regression Trees), a non-parametric statistical algorithm developed by Breiman et al. (1984), CART is capable of predicting or

analyzing both categorical (classification) and continuous or numerical (regression) data. Unlike other statistical analysis procedures, CART illustrates the data in the form of a decision tree, where each node is split into two nodes. And that is why CART is referred to as binary recursive partitioning techniques.

The tree starts from the root node containing the data objects, which is split into two child nodes, depending on the splitting criterion for the variable selected from the group of independent or explanatory variables. The result of the split can be a terminal or leaf node, which implies that it cannot be split further, or a decision node which consist of instances to be divided again into two child nodes. This process is repeated until resulting child nodes are homogenous or the predefined number of instances at leaf nodes is reached (Caetanoa et. al, 2005). Decision tree building in CART comprises three stages. In the first stage a complete tree with maximum size is grown by recursive partitioning the data. In the second stage a group of nodes is pruned using cross validation and cost complexity. In the final stage, a predictive error is considered as a criterion to select the optimal tree.

Growing the CART decision tree

The tree building process begins by splitting the root node into two child nodes, the best split is obtained when the impurity function, which exists between the parent node and two child nodes, is minimized. The best split equation is given by:

$$\Delta (s,t) = i(t) - (p_L\, i(t_L) + P_R\, i(t_R)) \quad (3.6)$$

Where *s* is the split of the independent variable, *t* is the parent node, *i(t)* is the impurity of node *t*, P_L and P_R are number of instances going to left and right nodes, and $i(t_L)$ and $i(t_R)$ are impurities of the left and right nodes respectively (Caetanoa et. al, 2005). For a classification tree *i(t)* is computed using a different criteria such as the Gini Index, Entropy Index, and Towing rule, which determine the best split. For a regression tree used to predict numeric class variable, the Least-Square Deviation is used as an estimate of node impurity, which is given by the following formula:

$$R(t) = \frac{1}{N_{w(t)}} \sum_{i=1} w_i f_i (y_i - \bar{y}(t))^2 \quad (3.7)$$

Where $N_w(t)$ is the measure of the weighted number of objects in node t, w_i is the value of the weighing variable for instance i, f_i is the frequency variable, y_i is the value of the dependent variable, and $\bar{y}(t)$ is the mean of the instances in node t.

Pruning the CART decision tree

For a fully grown decision tree in the initial step of CART, the tree will have many leaf nodes fitting the data, which leads to over-fitting and thus, the prediction accuracy is low for new unseen instances. Therefore, pruning, which develops an optimal tree, is needed, by shedding off the branches of large sub trees. The pruning process develops a sequence of smaller trees and computes cost complexity for each tree, and based on the cost-complexity parameter, the pruning procedure identifies the optimal tree with the highest accuracy. The cost-complexity parameter $R\alpha$ is set forth a linear combination of tree complexity and cost associated with the tree. Complexity is given by the following equation:

$$R_\alpha = R(T) + \alpha \left|\overline{T}\right| \Leftrightarrow \alpha = \frac{R_\alpha - R(T)}{\left|\overline{T}\right|} \quad (3.8)$$

Where $R(T)$ is the re-substitution error, $|T|$ is the number of terminal or leaf nodes in the tree, and α is the cost complexity associated with the tree. The re-substitution error $R(T)$ in case of a regression tree, is given by expected squared error, and is computed by the following equation:

$$R(T) = \frac{1}{N} \sum_{i=1}^{n} (y_i - d(x_i))^2 \quad (3.9)$$

Where (y_i, x_i) is the learning sample and d is the numerical predictor. The value of the complexity parameter in the pruning usually lies between 0 and 1. The pruning procedure develops a group of trees using different values of complexity parameter, giving different sizes of tree. According to Brieman et al. (1984),

among a group of trees of different sizes, for a value of α, only one tree of smaller size has high accuracy.

3.2.3 Logistic Regression classification

Regression is a collection of statistical function fitting techniques. These techniques are categorized according to the form of the function being fit to the data. Linear regression for example, is useful for data with linear relations or applications for which a first-order approximation is adequate. There are many applications for which linear regression is not appropriate or optimal. Because the range of the linear model using linear regression for data with continuous outcomes in (0, 1) or binary outcomes may not be appropriate. Logistic regression (Cox, 1958), sometimes referred to as Logistic regression models, is an alternative regression technique naturally suited to categorical (or dichotomous) data. Logistic Regression fits an S-shaped curve to the data. Such a shape, often referred to as sigmoid, is difficult to describe with a linear equation for two reasons. First, the extremes do not follow a linear trend. Second, the errors are neither normally distributed nor constant across the entire range of data (Peng et al., 2001). Let *X,Y* be a dataset with a binary response or class variable, where *X* is a vector of *k* independent variables (x_1, x_2,…, x_k) for each case in *X* the response or dependent variable is either y_i=1 or y_i=0, the logistic model predicts the logit of *Y* from *X*. The logit is the natural logarithm (ln) of odds of *Y*, and odds are ratios of probabilities π of *Y* happening (i.e., a work activity exists in an individual's schedule a specific day) to probabilities (1 –π) of *Y* not happening (i.e., a work activity does not exists in an individual's schedule a specific day). The simple logistic model has the following form:

$$log\,it(Y) = ln\left(\frac{\pi}{1-\pi}\right) = \alpha + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_k x_k \qquad (3.10)$$

Where *ln* is the natural logarithm, π is the probability of the class variable *Y*=1, α is the *Y* intercept, and β_1, β_2,…, β_k are the regression coefficients. The probability (π) that the class variable Y=1 is computed by:

$$\pi(y=1) = \frac{e^{\alpha+\beta 1 x 1+\beta 2 x 2+\ldots+\beta k}}{1+e^{\alpha+\beta 1 x 1+\beta 2 x 2+\ldots+\beta k}} \quad (3.11)$$

The α and β_1, β_2,…, β_k are typically estimated by the maximum likelihood method, which is preferred over the weighted least squares approach according to Haberman (1978) and Schlesselman (1982). The maximum likelihood method is designed to maximize the likelihood of reproducing the data given the parameter estimates.

The Logit models were trained and generated using the Rattle package for R (Williams 2009). The model was then exported to PMML and the *decisionMaker* class is used in the FEATHERS framework to deploy PMML Logistic regression models. The derivation of decisions (action assignment rule) from Logistic regression is used for model scoring as discussed in Chapter 2.

3.3.4 Multi-target classifiers using Info-Fuzzy Networks

Multi-target classifiers

Most data mining techniques (decision trees, Association rules, Naïve Bayes, artificial neural networks, etc.) work under the assumption that a classification problem has one target objective or class variable. Such assumption subsumes that an instance in the dataset do not contain more than one class variable. The common assumption of most data mining algorithms (decision trees, Naïve Bayes, artificial neural networks, etc.) that a learning task has only one class variable is very restrictive (Last, 2011, Caruana, 1997 and Suzuki et. al, 2001). In many real world datasets data objects may be concurrently assigned multiple class variables related to multiple tasks. These class variables or objectives may be strongly related to each other, weakly related or completely unrelated. Examples, as discussed in Dietterich et al, (1995), include student's grades in several courses, symptoms and diagnoses of a given patient, etc. A simple solution to the multi-objective classification problem is to generate a separate

model for each objective using any single-objective classification method. However, using a multi-objective model may be much more comprehensible than a collection of single-objective models. The storage and maintenance of multiple models in non-stationary systems may become a tedious task (Last, 2002). In addition, the combination of several single-objective classifiers in a single model may increase the overall predictive performance (Caruana, 1993).

In the next section the multi-objective Info-Fuzzy Networks is discussed in details, thus to provide a unified framework for single-objective and multi-objective classification, an extended classification task is presented below which includes the following components (based on Last, 2004):

- $R = (C_1, \ldots, C_k)$ – a set of k attributes in the dataset $(k \geq 1)$.
- C – a non-empty subset of n candidate input features (variables), where $C \subset R$ and $|C| = n \geq 1$. The values of these features are usually known and can be used to predict the values of target attributes.
- O – a non-empty subset of m target (output) attributes, where $O \subset R$ and $|O| = m \geq 1$. This is a subset of attributes representing the variables to predict. Discrete output attributes are also called class dimensions. The extended classification task is developed to predict the values of all target attributes, based on the corresponding dependency subset.

Multi-target classification Info-Fuzzy networks

As shown in Last (2004), an m-target classification function is represented by a multi-target info-fuzzy network (M-IFN), where each terminal node is associated with the probability distributions of all target attributes. The M-IFN model is an extension of an Oblivious Read-Once Decision Graph (OODG) called information network (IN) (Last and Maimon, 2004). As in OODG, the information network uses the same input attribute across all nodes of a given layer (level). The input attributes are selected incrementally by the IN induction algorithm to maximize a global decrease in the conditional entropy of the target attribute. The IN algorithm uses a prepruning approach: such that when no attribute causes a statistically

significant decrease in the entropy, the network construction is stopped. As shown in Last and Maimon (2004), the IN algorithm produce much more compact models than other decision tree learning models, and at the same time preserve nearly similar level of classification accuracy.

M-IFN construction is an iterative process; at every iteration the algorithm utilizes the entire training set instances to select an input variable which maximizes the decrease in the total conditional entropy of all class dimensions. The conditional entropy decrease, also called conditional mutual information gain (Cover and Thomas, 1991) is a feature selection criterion in single-target and multi-target decision tree algorithms. The conditional entropy measures the degree of uncertainty of a variable Y given the values of other variables $X_1,\ldots, X_n$ and it is calculated, as shown in Cover and Thomas (1991), as:

$$H(Y \mid X_1,\ldots, X_n) = -\sum p(x_1,\ldots, x_n, y) \log p(y / x_1,\ldots, x_n) \tag{3.12}$$

The conditional mutual information (MI) of the class dimension Y_i and the input variable X_n given the features $X_1,\ldots, X_{n-1}$ is calculated by (Cover and Thomas, 1991):

$$MI(Y_i; X_n \mid X_1,\ldots, X_{n-1}) = H(Y_i \mid X_1,\ldots, X_{n-1}) - H(Y_i \mid X_1,\ldots, X_n) = \sum_{x1\in X_1,\ldots x_n\in X_n, y_i\in Y_i} P(x_1,\ldots x_n, y_i) \log \frac{P(y_i, x_n \mid x_1,\ldots, x_{n-1})}{P(y_i \mid x_1,\ldots x_{n-1}) P(x_n \mid x_1,\ldots x_{n-1})} \tag{3.13}$$

Each internal node in the last layer, in a M-IFN with n-1 layer, represents a conjunction of values on n-1 input variables $X_1,\ldots, X_{n-1}$, As a result, the conditional mutual information of a class dimension Y_i and an input variable X_n given the variables $X_1,\ldots, X_{n-1}$ *over* all terminal (leaf) nodes z in the last layer L_{n-1} can be calculated as follows:

$$MI(Y_i; X_n \mid X_1,\ldots, X_{n-1}) = \sum_{z\in L_{n-1}} MI(Y_i; X_n \mid z) \tag{3.14}$$

The M-IFN algorithm evaluates discrete and continuous variables in a different way. Hence, the conditional mutual information of each discrete input variable X_j and the class dimension Y_i given a terminal node z is calculated using the following formula:

$$MI(Y_i; X_j \mid z) = \sum_{x_j \in X_j, y_i \in Y_i} P(z, x_j, y_i) \log \frac{P(y_i, x_j \mid z)}{P(y_i \mid z) P(x_j \mid z)} \quad (3.15)$$

Where x_j and y_i are distinct values of variables X_j and Y_i respectively.

The M-IFN algorithm, utilize the Likelihood-Ratio Test to evaluate the actual capability of an internal node to decrease the conditional entropy of an output by splitting it on the values of a particular input variable. The Likelihood-Ratio Test is a general-purpose technique for testing the null hypothesis H0 that two random variables are statistically independent. The default significance level (*pvalue*) for rejecting H0 is set to 0.1%. If the likelihood-ratio statistic is significant for at least one class dimension, the algorithm marks the node z as “split” on the values of an input feature Xj. in the case of the work activity process model dataset, the split occurs using the work status “wstat” variable. This variable is the split for all nodes of a given layer.

As discussed by Last (2004) the M-IFN algorithm comprise the following theoretical properties:

- In a *m* class dimension in an *n*-input *m*-dimensional model *M*, the average conditional entropy is not greater than the average conditional entropy over *m* single-target models S_i *(i=1,…, m)* based on the same *n* input variables. This inequality is reinforced if the multi-target model M is trained over more variables than the single-target models.
- The input variable selected by the algorithm will minimize the joint conditional entropy of all classes, either if the class variable is mutually independent or totally independent on each other.

Figure 3.1 shows the multi-target info-fuzzy network construction algorithm from a set of input variables. As discussed above the multi-target info-fuzzy network classification method is designed to find an accurate model(s) for predicting the values of *m* equally important class dimensions.

Input:	The set D of training examples; the set R of candidate input features; the set O of class dimensions; the minimum significance level *sign* for splitting a network node (default: $sign = 0.1\%$).
Output:	A dependency subset I of input features and an info-fuzzy network. Each input feature has a corresponding hidden layer in the network.
Step 1	Initialize the info-fuzzy network (single root node representing all examples, no hidden layers, and a target layer for all values of the class dimensions). Initialize the set I of selected inputs as an empty set: $I = \varnothing$.
Step 2	While the number of layers $\|I\| < n$ (total number of candidate input features) **do**

Step 2.1	**For each** candidate input X_j / $X_j \in R$; $X_j \notin I$ **do**
	If X_j is continuous **then**
	find the best threshold splits of X_j over all class dimensions O
	Calculate the total conditional mutual information between X_j and the class dimensions O: $cond\_MI_j = \sum_{Y_i \in O} MI(Y_i; X_j / I)$
	End Do

Step 2.2	Find the candidate input X_{j*} maximizing $cond\_MI_j$

Step 2.3	**If** $cond\_MI_{j*} = 0$, **then**
	End Do.
	Else
	Expand the network by a new hidden layer associated with the feature X_{j*}, and add X_{j*} to the set I of input features $I = I \cap X_{j*}$.

Step 2.4	End Do
Step 3	Return the set of input features I and the network structure

Figure 3.1 Multi-target Info-Fuzzy Network Construction Algorithm (adapted from Last, 2004)

3.3.5 Illustrative Example

To illustrate using the classification methods at the decision steps in the work activity process model, an illustrative example is given in Figure 3.2. Hence, using the *DecisionMaker* class and PMML implementation in FEATALB, it is now

easier to experiment the process model with different classifiers. Discrete classifiers are employed at discrete choice decision models and continuous classification methods are use at continuous decision models. As shown in Figure 3.2, at each decision step relative classification methods are utilized and validation results are computed.

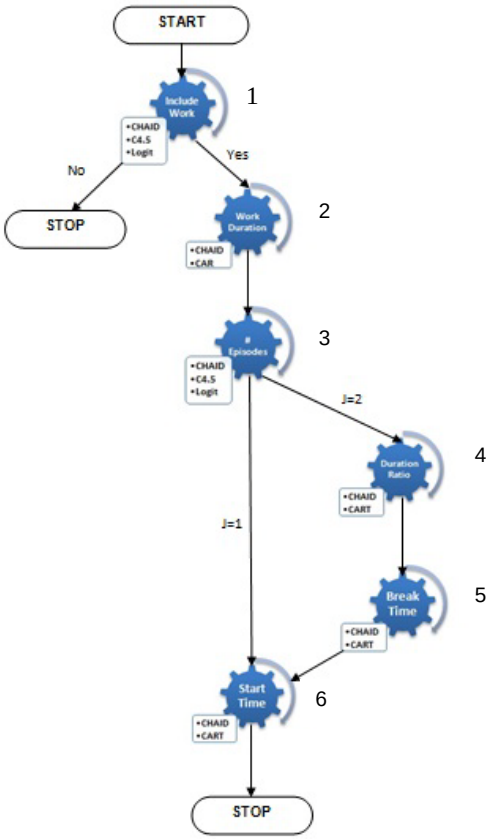


Figure 3.2 Employing discrete and continuous classification methods at decision steps

At discrete choice models (decision steps 1 & 3), are analysed using three classification methods, CHAID, C4.5, and Logit methods. On the other hand continuous models are trained and deployed using CHAID and CART decision tree methods. The reason for using CART instead of CHAID for some models is

that the latter is hard coded in ALBATROSS. As a result, training the models using different settings or modifying some model building parameters may be infeasible. Furthermore, changing the order of decision models will be plausible. The CHAID-based decision tree model is the original classification method used in ALBATROSS. Hence, when using CHAID for all decision models, the model will be referred to as the CHAID model. Consequently, when using the C4.5 and the Logit classifiers for discrete decision models, will be referred to as the C4.5 and Logit models, while maintaining the CHAID continuous decision trees for continuous decision models.

However, for the M-IFN multi-target classification method, the approach is different. All six decision steps (work, duration, number_of_episodes, ratio, break_time, and start_time) are predicted in one step. Using this approach all models are predicted without any preference and with equal priority. And this allows experimenting and analyzing the attributes interdependency and the activation dependency features in the process models. It is important to note that all decision steps in the process model share the same features.

3.4 Derivation of Decisions from Classifiers (Action Assignment Rules)

3.4.1 Derivation of Decisions from decision trees

The original induction method used in ALBATROSS is the CHIAD based induction method. Each discrete choice decision in the work activity process model is controlled by a decision tree model. Each decision model is derived from corresponding observations (training data set) in the activity diary data set. This section considers the action assignment rule from induction methods used to determine decisions in the prediction stage, as explained in Arentze and Timmermans (2005). Discrete and continuous choices are separately discussed.

3.4.1.1 Discrete Choice

In ALBATROSS the derivation of decision rules employs a probability distribution among classes. The levels at which decisions in the work activity pattern are taken which decides on the inclusion of work activity and the number of work episodes in the schedule. Accordingly, the definition of a case differs between decision trees. For example, the conceptual design is assumed that at the given moment in the decision process, a decision is derived for *N* cases. ALBATROSS employs a probabilistic action assignment rule, and this rule derivation method is used for the CHAID decision tree induction and C4.5 decision tree methods. A model in general and a decision tree in specific define a classification function as follows.

$$Pr\,(k|X_j) = f(X_j) \tag{3.16}$$

Where *k* is the leaf node index and X_j is a vector of attribute level for a given case j. Given that the decision tree model design is crisp and deterministic, the probability of assigning case j to node k is either 1 or 0 (Arentze and Timmermans, 2005). Subsequently, the action assignment rule using equation (3.1) becomes:

$$Pr\,(i|k) = f(q_k, \delta_j) \tag{3.17}$$

Where i is the index of discrete choice alternatives in the decision tree, q_k is the choice probability distribution of the class alternatives at leaf node k and δ_j is a 0-1 vector indicating the availability of each class in case j. It is important to note that q_k is a feature of the decision tree, while δ_j is calculated for each case during the prediction stage. Therefore, the probability of selecting alternative I in case j is calculated as follows:

$$Pr_j\,(i) = \sum_k Pr\,(k \mid X_j)\, Pr(i \mid k) \tag{3.18}$$

The probabilistic action assignment rule $f(q_k, \delta_j)$ used in ALBATROSS is calculated considering the subset of cases assigned to leaf node k, and for the sake of simplicity dropping the subscript k, can be written as:

$$P_{ij} = \delta_{ij}\left(\frac{q_i}{\sum_i \delta_{ij} q_i}\right) \tag{3.19}$$

Where p_{ij} is the probability of selecting choice alternative *i* in case *j* at leaf node *k*, δ_j is a 0-1 variable representing the availability of *i* in case *j*, and finally q_i is the probability of choice alternative i given by the decision tree at leaf node k and estimated on the training data.

3.4.1.2 Continuous Choices

The work activity pattern process model contains four continuous choice decision models, related to formulate and describing the work activity duration, start time,, the ratio between two work episodes, and the break time between two work episodes (if any). In ALBATROSS, continuous choice decision trees provide a specific distribution of the continuous dependent variable at each leaf node (Arentze and Timmermans, 2005). Consequently the continuous choice action assignment rule in accordance to equation (3.4) becomes:

$$Pr\,(y|k) = f(R_k, B_j), \quad y = 0, 1, 2, \ldots, 1440 \tag{3.20}$$

Where $Pr\,(y|k)$ is the probability of selecting value *y* at leaf node *k*, R_k is a vector containing parameters defining the distribution of values at leaf node k and B_j is a set of tuples in the form *(b1, b2)* containing blocked ranges [*b1, b2*] on dimension *y* in case *j* due to temporal constraints. ALBATROSS uses minutes as a measure of time and further the schedule has a time window of 24 hours, which implies that the value of y must have predefined minimum and maximum values. In addition y assumed to have natural numbers.

Continuous decision trees used in ALBATROSS define distributions at each leaf node specifying *m-1* cutoff points and the minimum and maximum of the range. The cutoff points divide the range into *m* intervals, taking in account that an equal number of training cases at leaf nodes is observed in each interval, this also means that R_k stipulate *m*+1 parameters. If the complete range of the schedule is available the number of elements of the set *Bj* is zero, if parts of the range are blocked by temporal constraints the range will be set to a value bigger than zero.

ALBATROSS employs a probabilistic approach for the continuous action assignment rule, to illustrate the method, consider the following notations. Let P_j (*y*) denote the probability of selecting *y* = 0, 1,…, 1440 in case *j, m* denote the number of Equal Frequency Intervals (EFI) used in continuous decision trees, d_i represent the width of equal frequency interval *i*, b_{ij} be the width of the blocked part of equal frequency interval *i* in case *j* defined by the combination of R_k and B_j and P_j (*y*) =1, if *y* falls in the unblocked part of the interval *I* and 0 otherwise.

$$Pj(y) = \sum_i Pr(i) Pr(y \mid i) \quad \forall j \tag{3.20}$$

Where Pr (*i*) is the probability of selecting EFI *i* and Pr (*y*|*i*) is the probability of selecting *y* given *i*. *Pr (i)* is defined as:

$$pr(i) = \frac{1}{m} \frac{d_i - b_i}{d_i} C_i \quad \forall i \tag{3.21}$$

The first term represents the priory probability of selecting *i*. Since EFIs represent an equal number of cases, an equal probability is assumed for all *m* EFIs. The second and third terms define a correction to this equal probability. The first correction is equal to the proportion of the available range in the EFI *i* and the third factor makes sure that probabilities sum up to one across EFIs.

Continuous decision tree describes a continuous distribution of responses for each leaf node. Figure 3.2 presents a graphical illustration of a continuous decision tree. The hypothetic example describes a tree that defines two-way splits on variables X1, X2 (in the ‘X1 = A’ branch) and X3 (in the ‘X1 = B’ branch). Associated with each of the *K* = 4 leaf nodes is a distribution representing the

responses on the response variable found in the corresponding partition of the training set. An action assignment rule defines for each new case at leaf node *k* how a prediction is drawn from a response distribution at leaf node *k*. Response distributions at leaf nodes can be represented in different ways. First, in case the distributions follow a standard normal form the mean and standard deviation would fully describe the distribution. However, durations and start-times of activities tend to deviate strongly from the normal distribution in the context of activity patterns (Greaves and Stopher 2000).

Other standard forms, such as for example a Poisson distribution, may approximate empirical distributions in some cases. Still, however, any standard form would only approximate with varying degrees of fit the empirical form. It is for this reason that we propose an assumption-free method for describing distributions at leaf nodes. Similar to the approach chosen by Greaves and Stopher (2000), this alternative way of describing a distribution is based on discretisizing a continuous range using the equal-frequency-interval method. In this method, the observed minimum and maximum (of training cases) at the leaf node defines the range of the variable. This range is then divided into *m* intervals by identifying the $m - 1$ cut-off points such that each interval includes the same number of observations. Hence, the method uses $m + 1$ parameters to describe each distribution at a leaf node including a minimum, maximum and $m - 1$ cut-off points, where *m* is a value set by the user. The higher the value for *m* the better the observed distribution is approximated. However, one may expect that at some point, model generalization will decrease with a further increase of *m*. Thus, the choice of *m* has a certain optimum, depending on sample size.

The FEATLAB framework uses the same approach of derivation of rules from decision trees i.e. the probabilistic action assignment rules as described below.

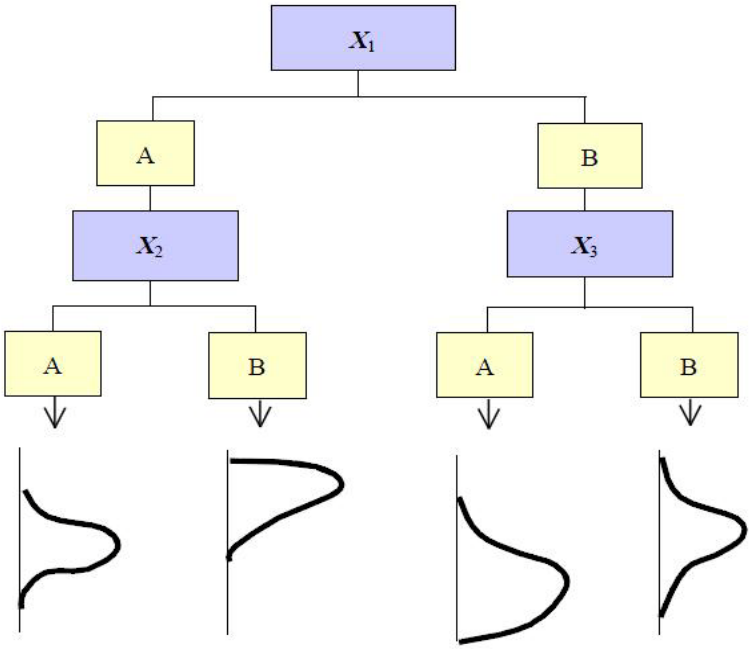


Figure 3.3 Graphical illustration of a continuous decision adapted from (Arentze and Timmermans, 2003)

3.4.2 Derivation of Decisions from Logistic regression models

The derivation of rules from logistic regression models as proposed by the literature of classification for logistic regression is using the probability equation. The probability (π) that the class variable Y=1 is computed by:

$$\pi(y=1)=\frac{e^{\alpha+\beta 1x1+\beta 2x2+...+\beta k}}{1+e^{\alpha+\beta 1x1+\beta 2x2+...+\beta k}} \quad (3.22)$$

The logistic regression equation attempts to model probabilities for the class variable Y, if the probability of a case is greater than a cutoff, usually for logistic regression the cutoff = 0.5, the model predicts Y=1. And if the probability is less than 0.5 the model classifies Y=0. Different cutoff values, other than 0.5 have

been proposed by the literature. However, most practical cases and without any supplementary information, such as the relative cost of misclassification or relative frequency of the class variable choice alternatives in the training data, 0.5 is recommended as a probability cutoff value. In FEATALB, the probabilistic action assignment rule is used, however, considering only the probability distribution obtained by equation 3.22. Consequently, for a specific case, $\pi (y=1)$ is the probability of predicting the positive class and $1- \pi (y=1)$ is the probability of predicting the negative class.

3.4.3 Derivation of Decisions from Multi-Target Info Fuzzy Network (M-IFN) Model

M-IFN generates the model as a set of rules which are responsible for predicting each class. For discrete class variables, the probabilities class alternatives or intervals are also specified. While for continuous models the average values of all cases belonging to each rule serves as the predicted value. Therefore and for model comparison purposes, the action assignment rules for discrete and continuous models adapted in decision tree models are also applied in the M-IFN models.

3.5 Model Comparison criteria and Model Validation

The model comparison criteria and model Validation are performed at three levels, the individual classifier (decision model) level, the activity pattern level, and the spatial and temporal levels. The reason for using these validation levels is that at each level provides more understanding of the model. For example at the individual classifier, the validation results provide information about the predictive performance about the classification method. Nevertheless, it does not give insight about the significance of the predicted schedules and the spatial-temporal resolutions. Validating the models at the activity pattern levels will be presented using the SAM method. This measure provides insight about the predicted activity patterns and how close they are to the observed ones. The validation at this level confirms the predictive performance at an aggregated

level. Finally, the validation at the spatial resolution (i.e. Origin-Destination (OD) matrices) and temporal level (the activity start time) provides aggregated validation information at the zonal and temporal resolutions.

3.5.1 Classifier Level Accuracy Analysis

3.5.1.1 Discrete choice models

The evaluation criteria for discrete choice models are presented using two accuracy measures. The first is the confusion matrix (also called contingency table) accuracy measure, since both discrete choice classifiers are binary. And the second measure is the Brier score (Brier 1950), because of the probabilistic action assignment rule used in scoring the models.

Confusion Matrix Accuracy Statistics

The confusion matrix records correctly and incorrectly recognized examples for each class. The following accuracy statistics can be derived from the confusion matrix:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{3.23}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{3.24}$$

$$Specificity = \frac{TN}{TN + FP} \tag{3.25}$$

$$F - Measure = \frac{2 \times Sensitivity \times Pr ecision}{Sensitivity + Pr ecision} \tag{3.26}$$

Where TP is the number of true positive values, FP is the number of false positive values, TN is the number of true negative values and FN is the number of false negative values. The precision in the F-Measure can be computed as: *precision* = *TP*/ (*TP+FP*). Accuracy is not a preferred performance measure for imbalanced datasets (Lim et al. 2000). Working with a highly imbalanced dataset, a classifier classifying everything as a majority class sample will result in a high

predictive accuracy. Sensitivity approximates the probability of the positive class being correctly classified, and specificity estimates the probability of correctly predicting the negative class. The F-measure focuses more on the dropout class taking into consideration sensitivity and precision as this measure is the weighted average of the precision and the sensitivity. An F-measure value reaches its best value at 1 and its worst value at 0.

Brier Score

The Brier Score (BS) is a metric related to the mean-squared-error often used in statistical fitting as a measure of model goodness. It is a descriptive measure often used in the literature on prediction accuracy. The Brier score is calculated as follows:

$$BS = \frac{1}{N}\sum_{i=1}^{N}\left(p_i - o_i\right)^2 \tag{3.27}$$

Where p_i is the predicted probability and o_i is the observed value of the instance i (0 if negative and 1 if positive) and N is the number of cases. The BS measures the average squared deviation between predicted probabilities for a set of events and their outcomes. So, a lower score represents a higher accuracy.

Taking in account that the derivation of decisions in discrete choice models is a probabilistic approach in the ALBATROSS/FEATHERS framework, as will be discussed later in this chapter, the BS is a sensible measure at the classifier level. In addition, the BS is not derived from the confusion matrix accuracy statistics hence it can be used to further confirm model performance and comparison.

3.5.1.2 Continuous models

The continuous choice models which were trained using only the CHAID tree induction method used originally in ALBATROSS were kept the same for the analyses performed using alternative discrete choice models. The performance of continuous choice models was assessed by means of the Relative Absolute

Error (RAE) which gives an indication of how good a predicted value is relative to the observed value. The reason for selecting this measure is that it can be reported as a percent error measure for numeric or continuous predictions. The RAE is calculated by dividing the sum of the absolute difference between the predicted and observed values by the observed cases.

Mathematically, the relative absolute error E_i of an individual program *i* is evaluated by the equation:

$$RAE = \frac{\sum_{i=1}^{n} |Pi - Ai|}{\sum_{i=1}^{n} |\overline{A} - Ai|} \tag{3.28}$$

Where P_i is the predicted value for case *i*, A_i is the actual value of case *i*, $\overline{A}$ is the mean of actual values, and n is the number of cases.

3.5.2 Activity Pattern Level

The work related to sequential analysis of activity patterns in activity-based models reached a new milestone, with the introduction of Sequence Alignment Methods (SAM) which was examined in transportation research by Wilson (1998). The predicted activity patterns can be compared to the observed ones by calculating a distance measure. This distance is based on the Sequence Alignment Method (SAM) and is obtained by calculating the effort required to make the two sequences identical. The interesting characteristic of the SAM is that it makes use of biological distance rather than geometric (Euclidean) distance as the basic concept of comparison (Joh et al. 2002). In Activity-based models, the SAM is used as a measure of goodness of fit at the activity pattern level (Arentze and Timmermans, 2000c; Moons, 2005; Janssens et al., 2006; Vanhulsel et al., 2007). The lower the SAM measure, (less operations of inserting, deleting or substitution of activities) the more similar the two sequences are.

Since experiments are conducted on the work process model only, the SAM distance is calculated for both, all activities in the schedule, and for work

activities only. The work activity patterns are expected to contain sequences of one, two or four symbols. It contains one symbol when the schedule does not contain a work activity so the sequence contains only a Home activity (e.g. H). It contains two symbols when the schedule a work activity with one episode (e.g. H W). Finally, the work activity pattern will contain four symbols when the schedule includes a work activity with two work episodes with a Break time in between (e.g. H W B W). The minimum number of symbols in the all activities sequence may be one, when the schedule does not contain any activities, i.e. the person under consideration stays at home on that day (H). On the other hand the maximum number of symbols for the all activities sequence is 11, i.e. when the person's schedule contains a work activity with two work episodes and he conducts all 7 activities considered in FEATALB.

The SAM method is fully implemented in the FEATALB framework. While running the simulation for model validation the SAM class methods are invoked for calculating the distance between predicted and observed activity sequences. The validation at the activity pattern level provides additional understanding and model assessment especially, the attribute interdependencies between decision steps feature. One of the outputs of FEATALB is a sequence of activities performed by an individual in a specific day. The order of activities that are performed is determined by the sequence of decision steps in the process model. The order of execution of decision steps in the process model is determined depending on the decision outcome. Therefore, validating the models at the activity pattern level using SAM confirms the activation dependency and attributes interdependencies in the scheduler process model.

The SAM distance is not just influenced by the difference in symbols but also by the length of the sequence, i.e. the number of symbols (activities) in the activity sequence. Thus, to be able to interpret the validation at the activity pattern level, a confusion matrix of work activity sequence lengths is created.

As discussed above the work activity sequence may contain one, two, or four symbols and thus, a confusion matrix of the actual versus the predicted sequence lengths are plotted in a confusion matrix.

3.5.3 Work Activity Trip Matrix and Trips Start Time Level Accuracy Analysis (Spatial and Temporal Resolutions)

At the trip matrix level, the observed and predicted Origin-Destination (OD) matrices are compared. A trip is the basic unit for calculating an OD matrix. An OD matrix contains the frequency of trips between an origin (row) and a destination (column). Based on the assumption that a trip starts from home and ends at home within a 24-hour time frame, symmetry of OD matrices is assumed. The zoning system inside FEATHERS is defined by three hierarchical geographical layers. This hierarchy contains land use data available at different level of detail. The Superzone level refers to all municipalities inside Flanders, while each Superzone is divided into Zones. Zones corresponds to administrative units (a zone belongs to only one Superzone) and similarly, each Zone is divided into Subzones. The Subzones level consists of virtual areas that are based on homogenous characteristics. There are a total of 327 Superzones, 1145 Zones, and 2386 Subzones.

In FEATHERS, OD matrices can be calculated at all of the three levels, where additional matrices, other than trip frequencies are generated. Additional travel information is aggregated in segmented OD matrices, such as ODs per each hour of the day, per transport mode, per activity type, ODs for each hour of the day per transport mode, per gender, per activity type, per age group etc.

To validate the models at the trip OD matrix level, the frequency of work activity trips for destinations at the Zone level in Flanders is aggregated forming a one dimensional array (1145 cells) with work activity trip counts at each zone for each day. The work trips at destination zones were selected and aggregated. Destination zones were chosen because it is a predicted value, keeping in mind that the source (i.e. the home location) for a person is always the same. Furthermore, ODs for all days of the week are aggregated and the correlation coefficient is calculated between observed and predicted OD matrix entries ρ(observed, predicted) to measure the degree of correspondence. It is important to note that work trips at destination zones was selected because it provides a level of detail more than the Superzone level on one hand, and a reasonable

aggregation level than the Subzone level, given the size of the cases in the data sets.

At the work activity start time (temporal) level, the work activities start times for each hour of the day is reported. The correlation coefficient between predicted and observed number of work activities per hour is calculated.

An advantage of using the correlation coefficient is that it is insensitive to the difference in scale between column frequencies (i.e. the difference in the total number of trips).

3.6 Attribute selection and discretization methods

3.6.1 Feature Selection: Relief-F

There are two main classes in feature selection techniques: the filter and the wrapper approach. The difference between them is the evaluation criterion used to select or rank attributes. For the wrapper approach, the ranking results from the estimation of the performance on the intended learning algorithm, while the filter approach evaluates features according to heuristics which is based on the characteristics of the data itself. Feature selection methods and their approaches have been compared comprehensively (Hall, 1999a, 1999b; Koller and Sahami, 1996). In this analysis, the filter approach, particularly the Relief-F feature selection method, is selected since it uses the data set characteristics to compute attributes relevance to the class variable.

The Relief feature selection (Kira and Rendall, 1992) is a distance-based technique weighting algorithm. It assigns an initial value of zero to each attribute that will be adapted with each run through the instances of the data set. The attributes with the highest weights are considered to be the most relevant, while attributes with weights close to zero or with negative weights are considered irrelevant. The weight of a particular attribute reflects its relevance in identifying the class attribute. Relief works by sampling instances randomly and finding its nearest neighbour from the same and opposite class. The nearest neighbour is

defined in terms of the Euclidean distance. Consider an *n-dimensional* sample space, defined by the variables $X_1,\ldots,X_n$, the distance between two instances *i* and *j* is calculated as follows:

$$d(i,j) = \sqrt{\sum_{k=1}^{n} (x_{ik} - x_{jk})^2} \quad (3.28)$$

Where $i = (x_{i1},\ldots,x_{in})$ and *j* are two dimensional vectors. The algorithm approximates the difference of probabilities for the weight of attribute x as follows:

$$Relief_X = P(\text{different value of } X \mid \text{different class}) - P(\text{different value of } X \mid \text{same class}) \quad (3.29)$$

An extension of the Relief feature selection technique that can handle multiple classes, noise caused by missing values, and outliers, is the Relief-F method developed by Kononenko, (1994). Relief-F attempts to find the k nearest hits and misses from each class and averages their contribution. The average is weighted by the prior probability of each class.

3.6.2 Discretization Methods

Discretization is an important data mining preprocessing task. Most machine learning algorithms are capable of extracting knowledge from data sets that store discrete attributes (Marzuki and Ahmad, 2007). However, many data sets contains continuous attributes, one solution to this problem is to make use of discretization methods which transforms them into discrete attributes. Discretization methods are used to divide the range of a continuous attribute into intervals (Kurgan. and Cios, 2001). In addition, discretization makes learning more accurate and faster (Dougherty et. al, 1995). The resulting model (decision tree, induction rules, etc.) are usually more compact and more accurate when compared to continuous models.

A typical discretization process consists of four steps (Liu et. al, 2002):

1. Sort all the continuous values of the attribute to be discretized.
2. Choose a cut point to split the continuous values into intervals.
3. Split or merge the intervals of continuous values.
4. Choose the stopping criteria of the discretization process.

In general, discretization methods are divided into two categories supervised and unsupervised methods. Supervised methods discretize attributes by taking into account the class attribute, while unsupervised discretization methods, that discretize attributes without taking into account the class labels.

In the work reported in this thesis, unsupervised discretization is used namely Equal Frequency Discretization (EFD). EFD is a simple unsupervised and univariate discretization method which discretizes continuous valued attributes by dividing the values into a specific number of intervals. Each interval contains approximately the same number of training instances, and each interval is associated with a distinct discrete value. In the FEATALB framework, the data sets contain continuous attributes which are discretized before training the models using EFD method. In addition, attribute interdependencies are preserved within decision steps, i.e. the inclusion of previous decision outcomes in the attribute list in subsequent decision steps. And thus, if the decision outcome is a continuous variable, it is discretized using EFD before included in the data set of subsequent decision steps. The data sets and types of attributes are discussed in details in the next chapter.

References

Arentze, T.A., Borgers, A., Hofman, F., Fujii, S., Joh, C., Kikuchi, A., Kitamura, R., Timmermans, H.J.P. and van der Waerden, P. (2000) Rule-based versus utility-maximizing models of activity-travel patterns. *Proceedings of the 9th international association for travel behaviour research conference*, Gold Coast, Queensland, Australia.

Arentze, T., Hofman, F., van Mourik, H., Timmermans, H. and Wets, G. (2000b) Using decision tree induction systems for modeling space-time behavior. Geographical Analysis, 32, 330-350.

Arentze, T.A., and Timmermans, H.J.P. ALBATROSS (2000): A Learning-based Transportation Oriented Simulation System. EIRASS, Eindhoven University of Technology, The Netherlands.

Arentze, T.A., and H.J.P. Timmermans (2003) Measuring the Goodness-of-Fit of Decision-Tree Models of Discrete and Continuous Activity-Travel Choice: Methods and Empirical Illustration, *Journal of Geographical systems*, 4,1-22.

Arentze, T. A., and H. J. P. Timmermans (2005). Albatross 2: A Learning-Based Transportation Oriented Simulation System. European Institute of Retailing and Services Studies, Eindhoven, Netherlands.

Brier, G. W. Verification of forecasts expressed in terms of probability. Mon. Wea. Rev., 1950, 78, 1–3.

Caruana, R. (1993). Multitask Learning: A Knowledge-Based Source of Inductive Bias. Proceedings of the 10th International Conference on Machine Learning, ML-93, University of Massachusetts, Amherst, pp. 41-48.

Caruana, R. (1997). Multitask Learning. Machine Learning, 28, pp. 41–75.

Cover T. M. & Thomas, J.A. (1991). Elements of Information Theory, Wiley.

Dietterich, T. G., Hild, H., & Bakiri, G. (1995). A Comparison of ID3 and Backpropagation for English Text-to speech Mapping. Machine Learning, 18 (1), pp. 51-80.

D. R. Cox. (1958) The regression analysis of binary sequences. Journal of the Royal Statistical Society, Series B (Methodological), 20(2): pp 215–242.

Dougherty, J.,Kohavi, R., and Sahami, M. (1995) Supervised and unsupervised discretization of continuous features. In Proc. Twelfth International Conference on Machine Learning. Los Altos, CA: Morgan Kaufmann, pp. 194–202.

Doherty, S. (2001) Classifying activities by time horizon using machine learning algorithms. Paper presented at the 80th Annual meeting of the Transportation Research Board, Washington, D.C., USA

GÄarling, T., BrÄannÄas, K., Garvill, J., Golledge, R.G., Gopal, S., Holm, E. and Lindberg, E. (1989) Household activity scheduling. Transport Policy management and Technology Towards 2001: Selected Proceedings of the 5th World Conference on Transport Research, 4, Western Periodicals, Ventura, CA, 235-248.

Greaves S, Stopher P (2000) Creating a synthetic household travel/activity survey – rationale and feasibility analysis. Paper presented at the 79th Annual Transportation Research Board Meeting, January 2000, Washington, DC, US.

Haberman, S. (1978). Analysis of qualitative data (Vol. 1). New York: Academic Press.

Hall M., Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009) The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

Hand, D. J. 1981. Discrimination and Classification. Chichester, U.K.: Wiley.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. Machine Learning, 11(1):63–90.

Janssens, D., G. Wets, T. Brijs, K. Vanhoof, T. A. Arentze, and H. J. P. Timmermans (2006) Integrating Bayesian Networks and Decision Trees in a Sequential Rule-Based Transportation Model. European Journal of Operational Research, Vol. 175, No. 1, pp. 16-34.

Joh, C-H, Arentze, T.A., Hofman, F. and Timmermans, H.J.P. (2002) Activity-travel pattern similarity: a multidimensional alignment method. Transportation Research B, 36, 385-403.

Kass, G.V. (1980) An exploratory technique for investigating large quantities of categorical data. Journal of the Royal Statistical Society. Series C (Applied Statistics) Vol. 29, No. 2, pp. 119-127.

Keuleers, B., G. Wets, T. Arentze, and H. Timmermans. Association Rules in Identification of Spatial-Temporal Patterns in Multiday Activity Diary Data. In Transportation Research Record: Journal of the Transportation Research Board, No. 1752, TRB, National Research Council, Washington, D.C., 2001, pp. 32–37.

King, G. and Zeng, L. (2001). Logistic regression in rare events data. Political Analysis, 9:137-163.

Kurgan. L and Cios, K.J.: Discretization Algorithm that Uses ClassAttribute Interdependence Maximization, Proceedings of the 2001 International Conference on Artificial Intelligence (IC-AI 2001), pp.980-987, Las Vegas, Nevada. (Pub. 2001.)

Last, M. (2002). Online Classification of Nonstationary Data Streams", Intelligent Data Analysis, Vol. 6, No. 2, pp. 129-147.

Last, M, Multi-objective Classification with Info-Fuzzy Networks. ;In Proceedings of ECML. 2004, 239-249.

Last, M, Vehicle Failure Prediction Using Warranty and Telematics Data, in Proceedings of the Next Generation Data Mining Summit: Ubiquitous Knowledge Discovery for Energy Management in Smart Grids and Intelligent Machine-to-Machine (M2M) Telematics, Athens, Greece, September 4, 2011.

Last, M and O. Maimon. A compact and accurate model for classification. IEEE Transactions on Knowledge and Data Engineering, 16(2):203–215, 2004.

Lim, T.S., Loh, W.Y. and Shih, Y.S. (2000) A Comparison of Prediction Accuracy, Complexity, and Training Time for Thirty-three Old and New Classification Algorithms" Machine Learning, 40, 203-228.

Liu, H. et. al: Discretization (2002) An Enabling Technique.Data Mining and Knowledge Discovery, 6,393-423.

Maimon O. & Last, M. (2000). Knowledge Discovery and Data Mining – The Info-Fuzzy Network (IFN) Methodology. Kluwer Academic Publishers, Massive Computing, Boston.

Moons, E. A. L. M. G, Wets, G., Aerts, M., Arentze, T.A. and Timmermans, H.J.P, 2004. The Impact of Simplification in a Sequential Rule-Based Model of Activity Scheduling Behavior, Forthcoming in Environment & Planning A

Moons, E. (2005) Modelling Activity-Diary Data: Complexity or Parsimony? PhD dissertation. Limburg University, Diepenbeek, Belgium.

Morgan, J.A. and Sonquist, J.N. (1963) Problems in the analysis of survey data, and a proposal. Journal of the American Statistical Association, 58, 415-434.

Pendyala, R.M., Kitamura, R. and Reddy, D.V.G.P. (1995) A rule-based activity-travel scheduling algorithm integrating neural networks of behavioral adaptation. Paper presented at the EIRASS Conference on Activity-Based Approaches, Eindhoven, The Netherlands.

Quinlan, J.R. (1986) Induction of Decision Trees, Machine Learning, vol. 11, no. 1, pp. 81–106.

Quinlan, J. R. (1993), C4.5: Programs for Machine Learning, Morgan Kaufmann.

Schlesselman, J. J. (1982). Case control studies: Design, control, analysis. New York: Oxford University Press

Suzuki, E., M. Gotoh, and Y. Choki (2001). Bloomy Decision Tree for Multi-objective Classification. L. De Raedt and A. Siebes (Eds.): PKDD 2001, LNAI 2168, pp.436 –447.

Timmermans, H.J.P., Arentze, T.A. and Joh, C-H. (2002) Analyzing space-time behavior: new approaches to old problems. Progress in Human Geography, 26, 175-190

Vanhulsel, M., Janssens, D., and Wets, G., 2007. Calibrating a new reinforcement learning mechanism for modeling dynamic activity-travel behavior and key events. CD-ROM. Proceedings of the 86th Annual Meeting of the Transportation Research Board, Washington D.C., U.S.A.

Weiss, S. I., and Kulikowski, C. 1991. Computer Systems That Learn: Classification and Prediction Methods from Statistics, Neural Networks, Machine Learning, and Expert Systems. San Francisco, Calif.: Morgan Kaufmann

Wets G, Vanhoof K, Arentze T A, Timmermans H J P. (2000) Identifying decision structures underlying activity patterns: an exploration of data mining algorithms" Transportation Research Record number 1718, 1 – 9.

Williams, G. J. Rattle: A Data Mining GUI for R. The R Journal, 1(2), 2009, 45-55

Wilson, C. (1998). Activity Pattern Analysis by Means of Sequence-Alignment Methods. Environment and Planning A, Vol. 30, , pp. 1017–1038.

Witten, I. H. and Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, second edition.

Z. Marzuki, F. Ahmad (2007) Data Mining Discretization Methods and Performances, Proceedings of the International Conference on Electrical Engineering and Informatics Institut Teknologi Bandung, Indonesia June 17-19.

Chapter 4

Flemish activity travel diary data

This chapter describes and discusses the data used in the analyses and the data sets employed to train and evaluate the models. The attributes types and interdependencies among data sets used for each decision step are further discussed. In addition, basic statistics and distribution of class variables are provided to gain better understanding the data. These statistics are used to assist the validation of individual models or choice facets in the process model. Activity pattern sequences of the observed schedules are generated. An activity pattern sequence consists of activities performed by an individual throughout the day. Activity sequences lengths are calculated to present the average number of symbols in an activity sequence. Finally Origin-Destination (OD) matrices for work activity are generated on the zone level.

In the next section the Flemish activity travel diary data is discussed and presented. In order to understand the data sets used to train the six models that constitute the process model by explaining condition and class variables.

4.1 Flemish Activity Travel Diary Data

The analysis and estimation of activity-based models requires data that are extracted from conventional travel surveys. Travel surveys contain information about the sequence of trips, performed by individuals, time of day, where, when, with whom, for how long, and with which transport mode. More precisely, data on activity patterns are required to build an activity-based model for transport demand. Contemporary household travel surveys mainly depend on the use of mail, telephone, internet and multimedia methods to obtain information on the daily travel and performed activities of a sample of the population (Kochan et. al, 2006).

Based on the discussion above, the data used to for building the activity-based model must provide measurements of activities at the end of trips and how and

when the survey respondent chose to perform them. The data sets used for training the models in FEATHERS originates from a travel survey for the Flemish study area. Hence, the Onderzoek VerplaatsingsGedrag Vlaanderen (OVG) survey is used. The OVG survey is a trip-based survey method, where information about trip purposes and information about activities in between trips are available. The OVG survey was conducted on 8,800 persons that were selected based on a random sample from the national register. The travel survey was primarily based on face-to-face interviews. In addition, information about the demographic, socioeconomic, household and trip-making characteristics of these individuals was collected.

The analyses reported in this thesis are restricted to the first component of the ALBATROSS model, i.e. the work activity model. For this reason, data sets for work activities were filtered out from the OVG travel survey. The filtering resulted in obtaining 5,288 cases (persons) in the *include work activity* data set as the first decision step in the work activity process model. Furthermore, the survey data was checked for data inconsistencies, specifically ensuring that schedules do not contain gaps in each individual's travel survey diary, which might originate from non-reporting of a trip while conducting the survey.

Table 4.1 shows the situational and socio-demographic variables that are used as prediction variables in FEATALB.

Household level attributes are urban density, household composition, presence of young children in the household, socio-economic class, and car ownership. Attributes related to the individual are gender, driver's license, work status and work status of the person's partner. Additionally, variables related to the measures of accessibility in decameters (dm) given the home location of the household (Xdag, Xn-dag, Xarb, Xpop, Ddag, Dn-dag, Darb and Dpop).

Table 4.2 depicts the class variables for each data set used in the decision models used in the process model. While executing the work activity process model these variables are predicted and included in the attribute set for the next decision model. Continuous variables such as duration, Ratio, Inter (break time duration) and start time are discretized using Equal Frequency Interval (EFI)

method. Tables 4.3 - 4.4 provides statistics for discrete choice and continuous class variables respectively. These data sets are used for building and validating the models that are used for predicting work activities in the process model. The datasets are divided using a 70-30% training-test split.

Name	Description	Categories
Urb	Urban density	0: highest density, 4: lowest density
Comp	Household composition	0:single without children, 1:single with children,2:single with parents,3: partner without children,4: partner with children
Child	Presence of the youngest children	0:no children, 1:< 6, 2: 6-12, 3: >12 years
Day	Day of the week	0: Monday to 6: Sunday
pAge	Age category	0: <35, 1: 35<55, 2: 55- <65, 3: 65-<75, 4:>75 years
SEC	Household income (in €)	0: <16,250, 1: 16,251 – 23,750, 2: 23,751 – 38,750, 3 >3: 38,750
Ncar	Number of cars in household	0: no cars, 1: 1 car, 2: 2 or more cars
Gend	Gender	0: female, 1: male
Driver	Driving license of person	0: is not a driver, 1: is driver
Wstat	Work status of person	0: no work, 1:part time, 2: full time
Pwstat	Work status of person's partner	0: no work, 1:part time, 2: full time
Xdag	Number employees daily-good sector within 3.1 km from home	0: <0,115], 1: <115,253], 2: <253,307], 3: <307,507], 4: <507,675], 5: >675
Xn-dag	Number employees non-daily-good sector within 4.4 km from home	0: <0,395], 1: <395,635], 2: <635,762], 3: <762,938], 4: <938,2525], 5: >2525
Xarb	Number employees within 4.4 km from home	0: <0,8785], 1: <8785,12995], 2: <12995,16120], 3: <16120,20199], 4: <20199,70314], 5: >70314
Xpop	Number households within 3.1 km from home	0: <0,5050], 1: <5050,8845], 2: <8845,13217], 3: <13217,16833], 4: <16833,22884], 5: >22884
Ddag	Distance (dm) to nearest 160 employees daily-good sector	0: <0,71], 1: <71,127], 2: <127,165], 3: <165,202], 4: <202,346], 5: >346
Dn-dag	Distance (dm) to nearest 260 employees non-daily-good sector	0: <0,92], 1: <92,145], 2: <145,176], 3: <176,258], 4: <258,334], 5: >334
Darb	Distance (dm) to nearest 4500 employees total	0: <0,92], 1: <92,128], 2: <128,201], 3: <201,274], 4: <274,360], 5: >360
Dpop	Distance (dm) to nearest 5200 households	0: <0,0], 1: <0,105], 2: <105,126], 3: <126,163], 4: <163,278], 5: >278

Table 4.1 Work activity data sets description

Name	Description	Categories
Work	Work	0:No, 1: Yes
Dur	Total duration (min.) of work activity	0:<0,395], 1:<395,495], 2:<495,526], 3:<526,565], 4: >565
More_Work_Ep	Number of work episodes	0: one, 1: two
Ratio	Ratio (%) between first and second work episodes.	0:<0,40],1:<40,48],2:<48,52], 3:<52,60], 4:>60
Inter	Duration (min.) of break time between first and second work episodes	0:<0,25], 1:<25,47], 2:<47,60], 3:<60,95],4:>95
StartTime	Work activity start time	0:<0,436], 1:< 436, 467], 2:< 467, 484], 3:< 484, 510], 4:< 510, 540] ,5: >540

Table 4.2 Class variables of models in the work activity process model

4.2 Basic Data Statistics and Distributions

The default work activity process model in FEAHTERS contains six decision steps, where decision steps *1* and *3* are discrete choice models. While decision steps *2*, *4*, *5*, and *6* are continuous models. As shown in Table 4.3, the include work model contains 5,288 cases, where 19 condition variables that are used in the models to predict the class variable. The class variable is a binary discrete variable which takes two values *0* for "no work" (nWo) and *1* for "work" (yWo). The include work data set is unbalanced with 27% towards the *yWo* class. The Number of work episode (More_Work_Ep) data set contains 1453 persons (i.e. persons who go to work). The data set contains 20 condition variables and the class variable is also a binary valued attribute *0* (one work episode) and *1* (two work episodes). Finally, the More_Work_Ep data set is highly skewed towards the negative class (one work episode) with 86%.

Data set	Number of condition variables	Number of cases	Minority class (%)
Include work	19	5288	27%
No. of work episodes (More_Work_Ep)	20	1453	14%

Table 4.3 Discrete choice models description

Table 4.4 illustrates a description of the continuous models, i.e. data sets for decision steps 2, 4, 5 and 6. The data sets contains 19, 20, 21, and 23 condition variables respectively, it is noteworthy that number of condition variables increase because of the attribute interdependencies between decision steps in the default process model. The average work duration is around 483 minutes however the standard deviation of the work duration variable is 129 which means that values are dispersed from the average work duration value as also depicted by Figure 4.1. The Duration ratio data set reports that the average ratio between the two work episodes is around 50 with a standard deviation of 13, which implies that in this data set the values are concentrated around the average as shown in Figure 4.2. The statistics for the Duration of break time model reveals that this data set is skewed to the left, as shown by Figure 4.3, with an average

break time duration of 70 minutes and a standard deviation of 60. In the context of activity-based travel demand modeling the statistics for this data set is rather realistic. Finally, the Work start time data set an average of 491 with a standard deviation of almost 60. This data set is also skewed to the left, i.e. work duration start times mostly start early as shown in Figure 4.4.

		Class variable			
Dataset	**Number of condition variables**	**Min.**	**Max.**	**Mean**	**Standard Deviation**
Work duration	19	75	800	483.5	129.2
Duration ratio	20	21	83	50.6	13
Duration of break time	21	10	300	70	60
Work activity start time	23	360	700	491.3	59.8

Table 4.4 Continuous choice models description

4.3 Activity Pattern and Origin-Destination (OD) Matrices statistics

An activity pattern for all individuals in the travel survey is generated for all activities in the schedule and for work activities only. Moreover, Origin-Destination (OD) matrices for trips initiated for work is also calculated on the zone geographical level.

Observed activity pattern sequences are extracted from the travel survey data for each individual moreover, observed work activity sequences are also extracted. Once a sequential activity-travel combination is extracted such as for instance Sleep-Eat-Work-Break-Work-Shop-Leisure-Sleep, it is meaningful to observe how activities are executed in time. Statistics about the observed activity pattern sequence are provided because the models will be validated at this level, using the Sequence Alignment Methods (SAM). The SAM is fully explained in Chapter 3.

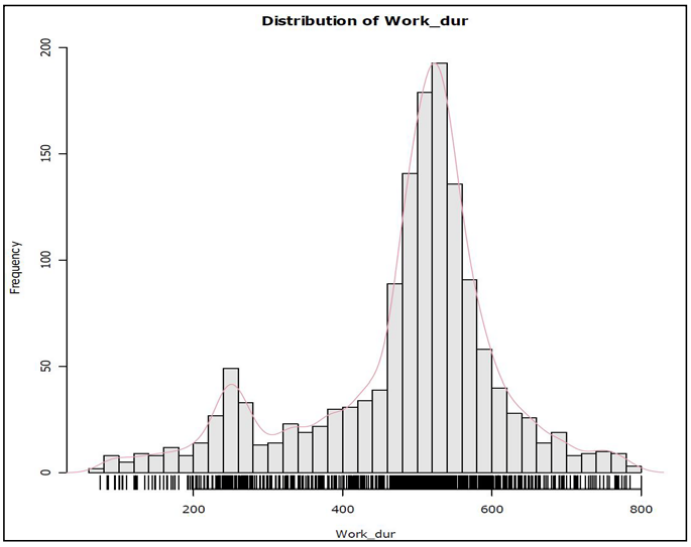


Figure 0.1 Distribution of the Work duration class variable

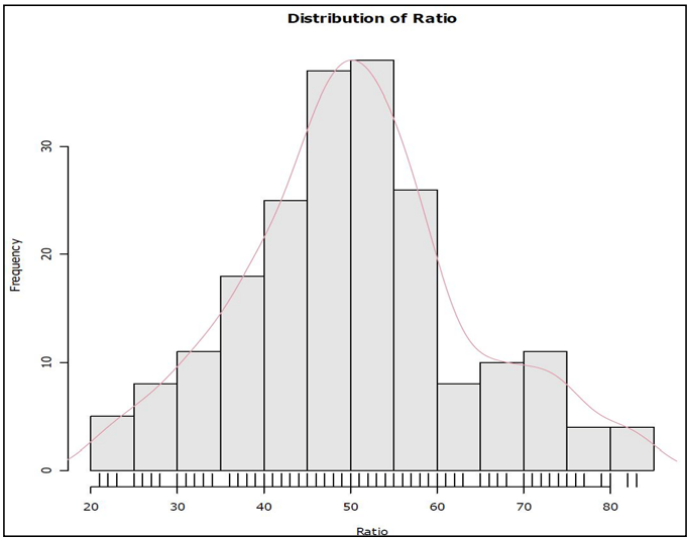


Figure 4.2 Distribution of the Ratio class variable

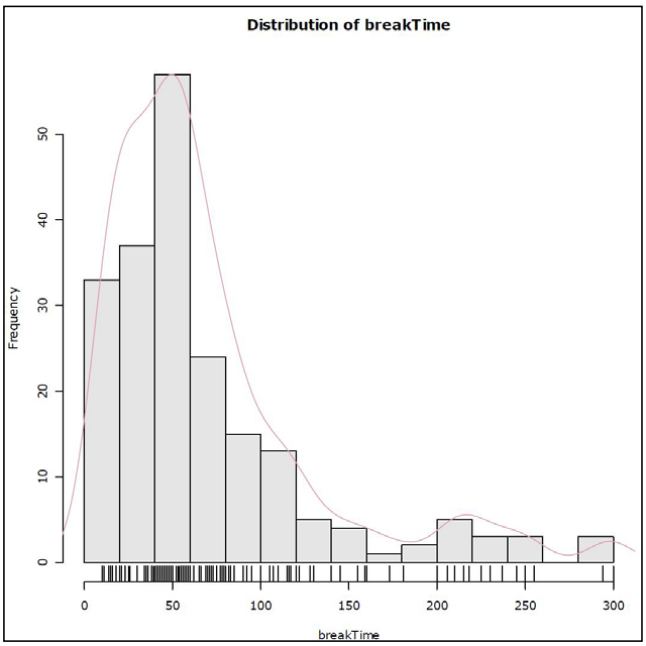


Figure 4.3 Distribution of the Work Break time class variable

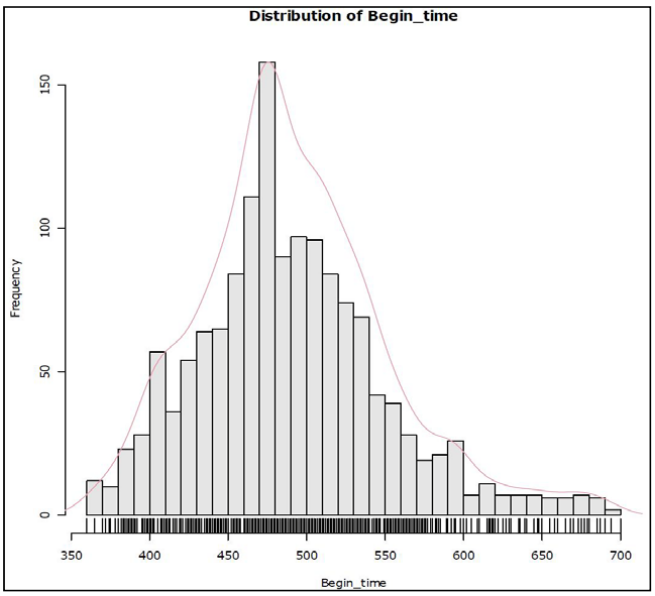


Figure 4.4 Distribution of the Work Begin time class variable

Table 4.5 shows statistics about the observed activity sequences and work activity sequences. In FEATHERS nine activities (Home, Work, Bring/get, Shopping, Services, Social visits, Leisure, Touring, and Other) are involved in the travel survey, where the Home activity is the default activity in all schedules for all persons.

	Average sequence Length	**Standard Deviation**
Observed Activity Pattern Sequence	5.16	2.3
Observed Work Activity Sequence	1.71	0.95

Table 4.5 Activity pattern sequence statistics

The average length of the observed activity sequence is around 5 symbols with standard deviation of 2.3, while the average length of the work activity sequence is almost 2 symbols with standard deviation of 0.95. Considering the work activity process model, the activity sequence may be one (Home), when the person's schedule contains no work activity, 2 symbols (Home-Work) when the schedule contains one work episode, and 4 (Home-Work-Break-Work), when there are two work episodes. These statistics will assess the validation of the models at the Activity pattern level.

The basic unit of calculating an OD matrix is the trip. An OD matrix constitutes of rows representing origins and columns representing destinations. In FEATHERS OD matrices are generated at three hierarchical levels, superzones, zones, and subzones. Statistics related for work trips OD generated at the zone level. The work activity is selected since the experiments are performed on the work activity process model. As shown in Table 4.6, for all 1145 zone 3422 work trips are performed with an average of 3 trips per zone.

	total Number of trips	Average number of work trips per zone
Work activity Trips	3422	2.99

Table 4.6 Origin-Destination work activity statistics

4.4 Conclusions

This chapter described the Flemish activity travel diary data and the data sets that were extracted from the survey, which are used in training the models (decision steps) in the work activity process models. Condition and class variables, for discrete choice and continuous models, are explained. In FEATHERS, continuous condition variables are discretized according to the Equal Frequency Discretization (EFD) method before the data sets are used for training the models. Moreover, to maintain attribute interdependencies model outcomes (predicted values) attributes are included as attribute in the data sets for subsequent decision steps or models. Continuous outcome variables are discretized using EFD before included in data sets of subsequent models.

The statistics of discrete choice models (Include work and number of work episodes models) revealed that they are unbalanced with 73% and 86% for decision steps 1 and 3 respectively. Continuous models statistics were also presented and revealed that the distributions are sensible.

Activity pattern and work activity sequences of the observed schedules are calculated. Statistics demonstrate that the average length of activities for the observed schedule is 5 symbols and around 2 symbols for work activities.

References

Kochan, B., Bellemans, T., Janssens, D., and Wets, G., 2006. Dynamic activity travel diary data collection using a GPS-enabled personal digital assistant. Proceedings of the Innovations in Travel Modeling Conference, Austin, Texas, U.S.A.

Part 2: Research Experiments and Results

In this part of the thesis the research design, experiments, and methodologies to achieve the aims are discussed. The objectives of this thesis is to firstly, study the effect of introducing new decision process models in ALBATROSS to the overall model performance. In addition, study the effect of modeling the decision steps in the process model simultaneously in one step. As a result, this might lead to finding better process models and to identify critical decision steps within the process model. Secondly, attempts to identify performance bounds for rule-based activity-based models. To accomplish these goals the work activity process model is elaborated in details. New work activity process models are introduced, additionally; the original CHAID-based decision tree is replaced by other induction methods for each new process model. Finally for each new process model the models are validated at three levels: (i) the induction method used at each decision step or choice facet level, using the predictive accuracy of each decision step in the scheduling process. (ii) The activity pattern level, Sequence Alignment Methods (SAM) is used to assess the correspondence between the observed and predicted activity patterns. (iii) The work activity trip matrix and trips start time level accuracy analysis (Spatial and Temporal Resolutions), where correlation coefficients are calculated to measure the degree of correspondence between the observed and the predicted Origin-Destination (OD) matrices and work activity start times.

Chapter 5

Research Design, Experiments, and Methodology

This chapter aims at introducing the research design and the experimental design presented in this thesis. In addition it discusses the methodology of the research to achieve the desired goals. The factors involved in improving process models performance are determined. To improve a process model performance one can improve quality of data used to train the models at individual decision step, obtain a better classifier, or find a better data representation within the process model. Moreover, a discussion on potential factors and the experiments required to improve the performance of process models are explained. The better classifier view is discussed by training and deploying the model using three approaches. Firstly, by modeling the decision steps in the process model simultaneously. This is achieved by using a multi-target classifier. Secondly, training and deploying models at each decision step independently i.e. without attributes interdependencies, which is referred to as the non-informed approach. And thirdly, by training the models while preserving attribute interdependencies and including the observed (actual) attribute values in subsequent decision steps, which will be referred to as the fully-informed approach. To asses the performance of process models, a performance lower (baseline model) and upper bounds are defined. Finally, the better data representation is discussed by explaining the work activity process model identifying critical decision steps and all possible execution paths.

5.1 Introduction

Computational process models constitute a powerful theoretical approach that conceptualizes choices as outcomes of using context-dependent heuristics. ALBATROSS, as a rule-based computational process activity-based model, consists of a series of agents that together handle the consistency of the data. The core of the ALBATROSS framework is the scheduling engine which controls

the scheduling processes as a sequence of decision steps. At each decision step the scheduling engine classifies the condition information for making a key decision. Hence, computational process rule-based models are based on a set of rules that represent transport choice behaviour. ALBATROSS employs a sequential decision process to generate daily activity schedules of individuals. The sequential decision process contains 26 decision steps, where at each decision step Chi-squared Automatic Interaction Detector (CHAID) based induction tree methods are utilized. However, a decision process containing 26 decision trees, where each decision tree contains many condition variables results in a complex process model.

Initiatives to investigate the complexity of the ALBATROSS decision process model have been undertaken. A study was conducted to investigate complex and simple classifiers within ALBATROSS by Moons, et al (2005). Simple models include OneR and Feature selection techniques. OneR is a very simple classifier that provides a rule that is based on the value of a single attribute (Holte, 1993), while feature selection techniques aims at reducing the number of irrelevant attributes, which as a consequence reduce the size of decision tree rules. On the other hand, complex models applied were C4.5 decision trees (Quinlan, 1986) and Support Vector Machines (SVM). The study showed that simple classifiers do not outperform complex models but are not inferior to complex models. However, the above mentioned studies were conducted on an earlier version of ALBATROSS where the scheduling process model contained only nine decision steps. In addition, some models obtained by very simple models may be insensible, as obtained in a study on the current version of ALBATROSS by Sammour et al. (2012). Therefore, obtaining a simple model that is sensible is needed. Janssens et. al, (2004) found that Bayesian networks performed better than CHAID decision trees in ALBATROSS. And that Bayesian networks are better suited to capture the complexity of the decision process model, since they take into account the interdependencies among the variables and decision steps outcome in the decision process model. Other classification methods, such as,

Association Rules were experimented with ALBATROSS as illustrated by Keuleers et al. (2001). In comparative studies by Wets, et al. (2000) and Moons, et al. (2004) revealed that different decision tree induction algorithms such as (CHAID, C4.5, CART etc.) achieve comparable results. However, investigations on modeling the decision steps, in the scheduling process model in FEATHERS/ALBATROSS, simultaneously in one model does not exist.

The analyses reported in this thesis are performed on the first component of the ALBATROSS model i.e. the work activity process model part of the scheduling engine. As discussed in Chapter 2, the work activity process model contains six decision steps. The first decision step evaluates whether the individual's schedule contains a work activity or not. If so, the duration of the work activity can be predicted. Next, the number of work activity episodes is predicted. If two work episodes are predicted, then the ratio between work episodes and the break time duration decision steps are executed. Finally, the work activity start time is predicted. Decision steps 1 and 3 are discrete choices, whereas, decision steps 2, 4, 5 and 6 are continuous models.

Based on the above discussion, the process model contains ***activation dependency***, since the output of some decision steps branches the execution. The first decision step evaluates whether the individual's schedule contains a work activity or not. If so, the duration of the work activity can be predicted. Next, the number of work activity episodes is predicted. If two work episodes are predicted, then the ratio between work episodes and the break time duration decision steps are executed. Finally, the work activity start time is predicted. It is noteworthy that if decision step 1 infers no work episode, then decision steps 2-6 will not be executed. Similarly, if decision step 3 evaluates to one work episode, then decision steps 4 and 5 will not be evaluated. A valid question arises here, is this sequence of decision steps is the best, or if the order of decision steps is changed will have an added value to the model?

To answer this question, alternative process models will be introduced and further analysed and validated in an attempt to obtain better sequential process model.

Additionally, ***attribute interdependency*** between decision steps is maintained, i.e. the inclusion of decision outcomes as attributes in subsequent decision steps. Figure 5.1 illustrates the activation dependency and attributes interdependency features in the work activity process model.

The dashed arrow lines indicate the inclusion of the output of a decision step as an input variable for the decision pointing out to. Continuous decision steps such as, work duration, Ratio, duration of break and start time are discretized using EFI before added to subsequent decision step(s). This functionality (***attribute interdependency***) also affects the performance of classifiers used in each decision step. Another question comes to mind, does this feature has an added value for the performance of the model? And to answer this question, an analysis of the work activity process model will be experimented with and without the attribute interdependency feature. This means that the models at decision steps will be trained and deployed with the inclusion of the actual values to replace the decision outcomes as attributes in subsequent decision steps. This will be referred to as the ***fully-informed*** approach. Conversely, models at decision steps will be trained independently (without including decision outcomes in subsequent decision steps) and this will be referred to as the ***non-informed*** approach. Additionally, modeling the decision steps in the work activity process model simultaneously will also be performed.

Another concern, aside from model complexity, is the validation of rule-based activity-based models. The data requirements for activity-based models are in general demanding compared to conventional travel demand models. This is obvious since this type of micro-simulation models should be able to predict the travel behaviour in detail. And so the validation of such behavioural models becomes a difficult task. In several experimental and analytical studies with ALBATROSS, e.g. the study of Janssens et al. (2006), Moons (2005), and Joh et

al. (2001), model validation has been performed at three levels; choice facet, activity pattern and trip matrix levels.

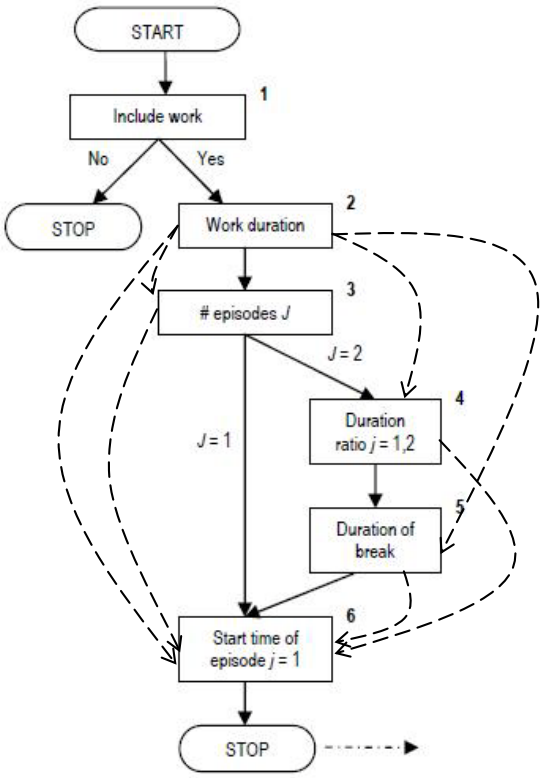


Figure 5.1 The work activity process model attributes interdependency diagram in ALBATROSS

In addition, a previous study by Sammour et al, (2012) proposed validation of FEATALB at the process model level by using SAM which provided extra understanding of the model by identifying critical decision steps. However, such validation levels indicate only the predictive performance of the models compared to observed data and/or comparing different classification methods. Hence, performance boundaries (lower and upper boundaries) for process models of rule-based activity-based models are required to assess and understand the complexity of such process models.

The rest of the chapter is structured as follows. In the next section the factors that improve process models performance are discussed and explained. Then the better classifiers view is explained in details by introducing multi-target classifiers, non-formed, and fully-informed approaches. Followed by the experimental design and performance bounds (Lower and upper process models performance bounds) section. The better data representation for improving process model performance is then discussed. And finally conclusions are presented.

5.2 Improving Process models

As discussed in the introduction of this chapter, the enhancement of the predictive performance of the work activity process model in FEATHERS / ALBATROSS may be achieved considering three major factors or views. Firstly, through obtaining better data originated from the travel diary data. Secondly, consider employing better classifiers using the attributes interdependency feature (fully-informed vs. non-informed approaches). Thirdly, consult a better data representation by the displacement of decision steps in the process model (the activation dependency feature).

The data sets used for training the models in the work activity process model, as well as the other 26 decision trees in FEATHERS, originate from the OVG survey. This survey is a trip-based survey. The travel survey was conducted based on a random sample taken from the national register. The travel survey was primarily based on face-to-face interviews. Furthermore, the random selection from the OVG survey is preprocessed and cleaned to assure schedule consistency (i.e. eliminating schedules that contain gaps). Additionally, considering the data sets that contain attributes that are related to the measures of accessibility given the home location of the household. These attributes provides location dependent information of the household and thus, cannot be improved. In view of the above discussion and the fact that data collection is expensive, investing efforts in the direction of obtaining better data is not feasible.

Several studies are conducted on improving process models by utilizing a better classifier at individual decision steps as discussed in the introduction of this chapter. Nevertheless, in the context of the FEATALB model, experiments conducted on an earlier version of the model, where the process model contained only nine decision steps and no activation dependency exist. Therefore, investigating the better classifier approach and experimenting with different classifiers (simple and complex) may improve the process model's performance.

As for the better data representation view, the current process model in ALBATROSS is based on the researchers experience in activity-based models. Thus, modifying the order of the decision steps in the process model may result in improving the predictive performance of process models. Introducing new decision sequences, i.e. changing the order in which the models are executed in the process model, may also improve the process model.

In the next sections, better classifier and better data representation as factors to improve the predictive performance are discussed in details.

5.3 Better Classifier View

As discussed in the introduction above, several studies have been conducted on investigating the predictive performance of rule-based activity-based models by comparing simple and complex classifiers at individual decision steps. Simple classifier models such as OneR and feature selection techniques, while complex classifiers include decision trees, SVM, Bayesian networks, Bagging and Boosting techniques, etc. However, the studies showed that simple classifiers do not outperform complex models but are not inferior to complex models (Moons, et. al, 2005). Moreover, the above mentioned studies were conducted on an earlier version of ALBATROSS where the scheduling process model contained only nine decision steps. Simple and complex classifiers will be investigated on the current ALBATROSS / FEATHERS framework using decision tree models and logistic regression at each decision step.

To this end, to investigate the predictive performance of rule-based activity-based models (the work activity process model component) the attributes interdependency between decision steps will be investigated. The models at each decision step will be trained in three different settings. First using the fully-informed approach, where actual values of decision outcomes are included in subsequent decision steps, as the process model executes. Second, using the non-informed approach, where the models are trained and deployed without the inclusion of decision outcomes to subsequent decision steps. Furthermore, the work process models will be modeled simultaneously in one model, using multi-target classifiers.

5.3.1 Multi-target classification Info-Fuzzy networks (M-IFN)

Instead of utilizing a separate classifier at each decision step, the work activity process model is modeled simultaneously. The M-IFN is chosen for the analysis because it was never used in modeling rule-based activity-based models. Using a multi-target model may be much more comprehensible than a collection of single-target models (Last, 2004). As proposed by the literature, the combination of several single-objective classifiers in a single model may increase the overall predictive performance (Caruana, 1993). However, multi-target models do not always increase predictive performance. As shown in a study by Piccart et al. (2008), that single-target models may be more accurate than multi-target models. As discussed in chapter 3, the M-IFN classification method makes use of the total conditional entropy of all class dimensions as the feature selection criterion. The conditional entropy is also used in single-target decision tree algorithms. The conditional entropy is used in the Information Gain and the Gain ratio, which are used as measures of attribute splitting in decision tree algorithms such as ID3 and C4.5 (Quinlan, 1986). Therefore, it is expected that the rules obtain by the M-IFN classifier will contain attributes that are similar to the first level of a decision tree model. It is also shown in several studies that decision tree models for rule-based activity-based models obtained comparable results.

Based on the above discussion, the M-IFN model is expected not to outperform decision tree models. Especially, using the M-IFN to model the work activity process model, the attribute interdependencies between decision steps is eliminated. Thus, the model (set of rules) obtained by M-IFN is expected to serve as a base line model (lower bound).

5.3.2 Set of Single target classifiers

In this section the training and the deployment of classifiers at individual decision steps of the work activity decision process is discussed. To investigate the added value of the attributes interdependency among decision steps the model is run in two settings. Firstly, the non-informed approach, in which the individual models are, trained excluding the attributes from previous decision steps. Secondly, the fully-informed approach, where the models are trained including the attributes (outcomes) of decision steps as attributes in subsequent models.

5.3.2.1 Non-informed Approach

In the non-informed approach the attributes interdependencies between decision steps are eliminated. In other words, the models at each decision step of the of the work activity process model are trained independently. To illustrate this, consider the model at decision step 3 (number of episodes) which includes 20 condition attributes and a discrete choice class variable. As shown in Figure 5.1 the outcome at decision step 2 (work duration) is discretized and then added as an attribute in the data set of this model (illustrated as a dashed line). In the non-informed approach all the arrowed dashed lines, which indicates the attribute dependency feature are removed.

5.3.2.2 Fully-informed Approach

In the fully-informed approach, the observed values rather than predicted outcomes are included as attributes in subsequent decision steps. In real situations, this model cannot be obtained. Nevertheless, it is expected that the

performance of the models will be increased. Since, the observed attributes values are included for each case. And thus, such performance boundary can serve as the upper performance bound for the model. Classifiers such as CHAID, C4.5 and Logistic Regression, will be used in the experiments in this approach. CHAID and C4.5 decision tree methods are used to further prove the similar performance and compare these results with those of the C4.5 models obtained in the non-informed approach. This also applies to the Logistic Regression models.

5.4 Experimental Design and Performance Bounds (Lower and upper process models performance bounds)

In order achieve the desired goal in analysing and improving process models, the experiments on the classifier and data representation views are conducted. However, to assess the performance of rule-based activity based models, the models are validated at three different levels: (1) The individual classifier level, using confusion matrix accuracy statistics, (2) the Activity pattern level using the SAM distance measure, and (3) the spatial-temporal resolution, by calculating the correlation coefficient between observed and predicted work trips at each zone, and work activity start times.

These validation levels assess the performance of a model compared to each other. Nonetheless, they do not provide information on how much a model's performance is superior to a base line model. Additionally, when the predictive performance of process models is experimented, it is difficult to appraise the added value of attributes interdependencies or activation dependencies. Therefore, a base line model (lower bound) needs to be defined such that the added value of using better models is assessed. Correspondingly, an upper performance bound as the best possibly achieved model is defined (fully-informed approach). These performance bounds allows one to evaluate and measure the added value of a model related to the base line model, attributes interdependencies, and activation dependencies of process models.

5.5 Data Representation View

The original work activity process model in ALBATROSS is replaced by other process models, i.e. new process models are introduced. This allows for evaluating which data representation (including preserving the attributes interdependencies feature) of the decision results in a better performance. The alternative process models are developed such that they produce the same information needed to output schedules.

The activation dependency in the work activity process model branches the execution of the decision steps. Depending on the outcome of some decision steps an execution path is obtained. Depending on the decision outcomes three execution paths exist. For example as shown in Figure 5.1 if at decision step 1 the model predicts no work activity (path 1) the process model terminates. And if the model predicts a work activity, then the duration (decision step 2) and the number of work episodes (decision step 3) are predicted. If one work episode is predicted, the work activity start time (decision step 6) is obtained (path 2: 1-2-3-6). If two work episodes are predicted, then the ratio model (decision step 4) and the break time duration (decision step5) are consulted. And finally the work activity start time (decision step 6) is predicted (path 3: 1-2-3-4-5-6).

The current sequence process model in ALBATROSS is based on expert's opinion. However, other logical process models may exist which may result in better process models. This implies that by representing the data in a different way (changing the decision steps sequence), might improve the predictive performance of process models.

5.6 Conclusion

In this chapter, the research methodologies, design of experiments and the performance bounds are explained. In addition a discussion on the factors that are involved in improving the predictive performance of process models. The first

factor is to obtain quality data, while the second factor using a better classifier and the third by attaining a better data representation, i.e. changing the sequence of decision steps in the process model.

The experiments to identify performance bounds will be conducted by training and deploying the work activity process model in three settings. The first setting is by training the models (decision steps) of the process model using a multi-target classification method. A multi-target classification method allows for predicting all the decision outcomes of the models in one step. Modeling the whole process model in one step will eliminate the attributes interdependency and the activation dependency features that exist in the process model. And will assess to understand the effects of these features on the predictive performance of the process model. The second setting (non-informed approach) is by eliminating the attributes interdependency feature, i.e. training and deploying the models without including decision outcomes to subsequent decision steps. This setting will help analyse the attributes interdependency feature on the predictive performance of the process model. Furthermore, comparing its performance with the multi-target's model performance will appraise the added value (if any) of the attributes interdependency feature on the model. The third setting (fully-informed approach), which is the ideal performance of the model, is by training and deploying the models at the decision steps by including the actual values to be included to subsequent decision steps. This setting will allow for setting the maximum performance possible (upper performance bound). And as a result will help measure the magnitude of predictive performance of the process model. This can be achieved by comparing the results of the three settings. In addition, the added value of the attributes interdependency and the activation interdependency features and their effects to the predictive performance of the process model can be measured.

The data representation alternatives in enhancing the predictive performance can be obtained by determining different process models. In other words, represent the decision models in a different order. Three logical data representations (process models) will be presented, analysed, and validated.

Model sensitivity terms of the parameters used to train decision tree models and the action assignment rules and their effect on the predictive performance will also be performed. Before training decision tree models, some parameters must be set before the models are constructed and deployed. One important parameter is the minimum number of cases at leaf nodes. This parameter influences the decision tree building and pruning. All decision tree models in ALBATROSS are trained be setting the minimum number of cases at leaf nodes to 30. This number worked well and achieved a desirable predictive performance for the Dutch data, but how about the Flemish data? For this reason the effect of increasing or decreasing this number will be analysed to assess the predictive performance of the process model. Action assignment rules involve the outcome of prediction or providing a decision in a decision tree. ALBATROSS makes use of probabilistic action assignment rules, i.e. at leaf nodes the distribution of the class variable is considered before supplying a decision. Nevertheless, using deterministic (crisp) action assignment rules will help identify the sensitivity of the models and assess the predictive performance of the process model.

In chapter 6, the work activity process model is experimented using different classifiers. This involves modeling the work activity process model simultaneously using a multi-target classifier, non-informed, informed, and the fully-informed approaches. Followed by chapter 7 where alternative process models are introduced and their performance and validations are explicated. Then in chapter 8 a sensitivity analysis of decision tree models used at individual decision steps is performed. Finally the thesis concludes with chapter 9 with the final discussion, final conclusions and future research.

References

Beau Piccart, Jan Struyf, and Hendrik Blockeel. (2008). Empirical Asymmetric Selective Transfer in Multi-objective Decision Trees. In Proceedings of the 11th International Conference on Discovery Science Pages 64 – 75.

Caruana, R. (1993). Multitask Learning: A Knowledge-Based Source of Inductive Bias. Proceedings of the 10th International Conference on Machine Learning, ML-93, University of Massachusetts, Amherst, pp. 41-48.

Holte, R. C. (1993). Very simple classification rules perform well on most commonly used datasets. Machine Learning, 11(1):63–90.

Janssens, D., Wets, G., Brijs, T., Vanhoof, K., Timmermans, H.J.P., and Arentze, T.A., 2004. Improving the performance of a multi-agent rule-based model for activity pattern decisions using Bayesian networks. Transportation Research Record, Vol. 1894, pp. 75-83.

Janssens, D., G. Wets, T. Brijs, K. Vanhoof, T. A. Arentze, and H. J. P. Timmermans (2006) Integrating Bayesian Networks and Decision Trees in a Sequential Rule-Based Transportation Model. European Journal of Operational Research, Vol. 175, No. 1, pp. 16-34.

Joh, C., H. Arentze, T.A. and Timmermans, H.J.P. (2001) Pattern Recognition in Complex Activity-Travel Patterns: A Comparison of Euclidean Distance, Signal Processing Theoretical, and Multidimensional Sequence Alignment Methods. Presented at the 80th Annual Meeting of the Transportation Research Board, Washington, D.C., USA.

Keuleers, B., G. Wets, T. Arentze, and H. Timmermans. Association Rules in Identification of Spatial-Temporal Patterns in Multiday Activity Diary Data. In Transportation Research Record: Journal of the Transportation Research Board, No. 1752, TRB, National Research Council, Washington, D.C., 2001, pp. 32–37.

M. Last, Multi-objective Classification with Info-Fuzzy Networks. ; In Proceedings of ECML. 2004, 239-249.

Moons, E. A. L. M. G, Wets, G., Aerts, M., Arentze, T.A. and Timmermans, H.J.P, 2004. The Impact of Simplification in a Sequential Rule-Based Model of Activity Scheduling Behavior, Forthcoming in Environment & Planning A.

Quinlan, J.R. (1986) Induction of Decision Trees, Machine Learning, vol. 11, no. 1, pp. 81–106.

Sammour G., Bellemans T., Vanhoof K., Janssens D., Kochan B., and Wets G. (2012) The Usefulness of Sequence Alignment Methods in Validation Rule-Based Forecasting Models. Transportation Journal, 39, 773-789.

Wets, G., Vanhoof, K., Arentze, T. A., Timmermans, H. J. P. (2000) Identifying Decision Structures Underlying Activity Patterns: An Exploration of Data Mining Algorithms Transportation Research Record, 1718, pp. 1-9.

Chapter 6

Upper and Lower Performance Bounds for Process models of Work activity process models

The work activity scheduling process in ALBATROSS is the first component that is responsible for generating primary work activities and their start time, duration of each work episode if more than one episode is predicted. It is a sequential process model that contains six decision steps, where at each decision step the system applies a set of rules (if-then rules). In ALBATROSS rules at each decision step are derived using a Chi-squared Automatic Interaction Detector (CHAID)-based induction method. This method is applied to generate decision trees trained from activity-travel diary data. This chapter aims identifying lower and upper performance bounds for process models used in rule-based activity-based models. The objectives are achieved by training and deploying the models in three approaches: (1) modeling the decision steps in one step using a multi-target classification method. (2) Training the models at decision steps preserving the attributes interdependencies among models (fully-informed approach) while including observed rather predicted decision outcomes in subsequent decision steps, and (3) training the models at decision steps while eliminating the attributes interdependencies (non-informed approach).
The multi-target classification method is the lower bound because it has the lowest fitting capacity. We consider the fully-informed approach as the upper bound because it has more information than the non-informed approach and it has the same fitting capacity.

6.1 Introduction

The work activity scheduling process model in FEATALB is an important component, because it is the first process model executed in the sequential decision process. The schematic diagram of the work activity process model, as depicted in Figure 2.1, shows that each numbered rectangle represents a

decision step. Each decision step is used to predict a specific model related to work activities. Prediction is performed according to a decision tree model that is trained and derived from activity-travel diary data. The process starts by deciding whether or not the individual's schedule contains a work activity, further if yes the duration of the work activity is then predicted. Followed by the number of work activity episodes, subsequently the ratio between work episodes and the break time duration is decided, and finally the work activity start time is predicted. Decision trees 1 and 3 are discrete choice decisions. On the other hand decision trees 2, 4, 5 and 6 are continuous choice decisions.

As discussed in previous chapters, the process model contains *activation dependency*, in view of the fact that the output of some decision steps branches the execution of the process model. In addition, the attributes interdependency between decision steps may influence the predictive performance of the model. Especially that the output of a decision step is included in the attribute list of subsequent decision steps. This may result in achieving better models if this new attribute is relevant and correlates to the class attribute.

Most data mining techniques (decision trees, association rules, naïve Bayes, artificial neural networks, etc.) work under the assumption that a classification problem has one target objective or class. Such assumption subsumes that an instance cannot belong to more than one class at the same time. The assumption of most data mining algorithms that a learning task has only one class is very restrictive (Caruana, 1997) (Suzuki, 2001). In many real world data sets data objects may be concurrently assigned multiple class labels related to multiple tasks (Last, 2004). These target objectives or classes may be strongly related to each other, weakly related or completely unrelated. Hence, classifying multiple classes or labels data sets can be performed using two approaches. The first approach is to generate a separate model for each objective/class using any single-target classification method. In the second approach we classify all cases using a multi-target classification method in one model.

The performance of activity-based models is usually measured either by validating a test set against a training set at various levels of the same model, or

by comparing accuracy statistics of different models. However, performance boundaries for rule-based activity-based models do not exist. Performance boundaries can be determined by identifying a lower (base line) and an upper performance bound. The analysis in this chapter aims at identifying performance bounds for process models of activity-based models. This is achieved by training the decision steps in the work activity process model in three settings. First, the decision steps will be trained and modeled simultaneously using the M-IFN classification method. Second, the models at individual decision steps are trained using the fully-informed approach i.e. including the real values of the decision outcomes as attributes for subsequent decision steps. Third, the models are trained using the non-informed approach i.e. without including the outcome of decision step as attributes in the data sets in subsequent decision steps.

The remainder of the chapter is structured as follows. In the next section the lower performance bound using the M-IFN model is explained. Then the fully-informed approach is discussed by training and deploying individual models using three classification methods (CHAID, C4.5, and Logistic regression) to identify the upper performance bound. In the fully-informed approach, the models are trained while including actual values of decision outcomes of decision steps as attributes in the datasets for subsequent models. Followed by the non-informed approach analysis, in which the models are trained without the inclusion of decision outcomes in subsequent models. The discussion of the validation results of the three approaches to identify lower and upper performance bounds then takes place. Finally the chapter ends with the conclusions.

6.2 Lower Bound: Multi-target Classification using Info-Fuzzy Network Methods (M-IFN)

A prediction problem for some learning tasks e.g. the work activity process model in FEATHER/ALBATROSS, assumes a sequential process of decision steps, where each decision step is a model or objective in its own. The basic solution to this problem is to use an individual classification method in each decision step in the sequence as already implemented in ALBATROSS. Consequently, in rule-

based activity based models, some components i.e. the work activity process model in ALBATROSS, constitute six highly related (attributes interdependencies) decision steps. The six decision steps contain models that are generated from activity-diary data set, in which they share many attributes. Therefore, in the work activity process model, while running the simulation after consulting one model in a decision step the output is captured and included in the attribute set of the next decision step. The resulting individual classification models may be the best for each individual target variable, but adapting a multi-target classification method in a single model may be intelligible (Last, 2002).

The M-IFN classification method utilizes the total conditional entropy to select the most relevant attribute(s) to be used in the generated rules for classification. Additionally as discussed in Chapter 3, the M-IFN method utilizes the *Likelihood-Ratio* Test to evaluate the actual capability of an internal node to decrease the conditional entropy of an output by splitting it on the values of a particular input variable. The Likelihood-Ratio Test is a general-purpose technique for testing the null hypothesis H0 that two random variables are statistically independent. The default significance level (pvalue) for rejecting H0 is set to 0.1%. If the likelihood-ratio statistic is significant for at least one class dimension, the algorithm marks the node z as "split" on the values of an input feature X_j.

The data sets used to train the models in individual decision steps in the work activity process model are integrated in one data set with six class (target) attributes (*Work, Work_Dur, More_Work_Ep, Ratio, Break_time, and Begin_Time*). The M-IFN model is trained and the rules generated using the IFN software developed by Last (2004). After training the model with IFN software, the model output is a set of rules (8 rules for the work activity process model) where all six variables are predicted according to the model.

The target variables have the following characteristics:

- *Work,* which takes the value of 1 if the individuals schedule contain a work activity and the value of 0 otherwise.
- *Work_Dur* presents the duration of the work activity in minutes.

- *More_Work_Ep,* which takes the value of 1 if the schedule contain two work episodes and the value of 0 if the schedule contain only one work episode.
- *Ratio* represents the ratio of the first work duration to the second work duration.
- *Break_time* is the duration of the break time (in minutes) between the first and second work episodes.
- *Begin_Time, which measures the begin time of the work activity.*

The model also selected two input variables:

- *wstat* (work status of the person), which takes the value of 0 if the person is unemployed and the value of 2 is the person is employed.
- *Day,* which represents the day of the week (0: Monday, 1: Tuesday …6: Sunday).

The M-IFN model prediction rules are listed below. The name of each discrete target attribute is followed by the probabilities of its values or intervals between parentheses.

1- IF (wstat = 0) THEN
Work = 0 (0.977, 0.023), Work_Dur = 422, More_Work_Ep = 0
(0.977, 0.023), Ratio = 41.8, Break_time = 95.8, Begin_Time=514

2- IF (wstat = 2 AND Day = 0) THEN
Work = 1 (0.35, 0.65), Work_Dur = 309.34, More_Work_Ep = 0
(0.893, 0.107), Ratio= 52.38, Break_time = 93.2, Begin_Time = 493.9

3- IF (wstat = 2 AND Day = 1) THEN
Work = 1 (0.335, 0.665), Work_Dur = 334.5, More_Work_Ep = 0
(0.925, 0.075), Ratio= 49.2, Break_time=52.4, Begin_Time=497.2

4- IF (wstat = 2 AND Day = 2) THEN
Work = 1 (0.355 / 0.645), Work_Dur = 304.6, More_Work_Ep = 0
(0.947, 0.053), Ratio= 51.5, Break_time= 63.4, Begin_Time= 482.5

5- IF (wstat = 2 AND Day = 3) THEN
Work = 1 (0.308 / 0.692), Work_Dur = 334.9, More_Work_Ep = 0
(0.89, 0.11), Ratio= 52.3, Break_time= 55.2, Begin_Time= 494.1

6- IF (wstat = 2 AND Day = 4) THEN
Work = 1 (0.428 / 0.572), Work_Dur = 278.33, More_Work_Ep = 0
(0.928, 0.072), Ratio= 50, Break_time= 66.1, Begin_Time= 483.1

7- IF (wstat = 2 AND Day = 5) THEN
Work = 0 (0.867/ 0.133), Work_Dur = 54.35, More_Work_Ep = 0
(0.987, 0.013), Ratio= 43, Break_time= 59.5, Begin_Time= 491

8- IF (wstat = 2 AND Day = 6) THEN
Work = 0 (0.934/ 0.066), Work_Dur = 28.79, More_Work_Ep = 0
(0.996, 0.004), Ratio= 30, Break_time= 20, Begin_Time= 475.7

The rules are simple and it is noted that the model is sensible and as will be illustrated later that the input variables used in the rule (i.e. *wstat* and *Day*) correspond to the decision tree models (CHAID and C4.5) up to the second level. As revealed by the model rules above, if the person is unemployed then no work episode will be predicted. If the person is employed, then the day attribute is evaluated. If it is a working day (*0 - 4*), then the model predicts a work activity otherwise no work activity is predicted. Considering the *More_Work_Ep* it is noted that model always predicts one work episode. This occurs because the *More_Work_Ep* is highly skewed (87%) towards the one work episode class (0). As for the rest target attributes (*Work_Dur, Ratio, Break_time, and Begin_Time*), which are continuous variables, the predicted values at each rule represents the average of the values to which the instances belong.

Since M-IFN provide the probability distribution for discrete choice target variables, as shown in the rules above, hence the probabilistic action assignment rules is used, as discussed in chapter 3. The probabilistic action assignment rule employs a stochastic approach in which the probability distribution is considered when predicting a class variable. Similarly, for continuous target variables the same approach as the one discussed in chapter 3, for continuous models is used. The action assignment rule for continuous variables uses the distribution representing the responses on the response variable found in the corresponding partition of the training set. For more on action assignment rules for discrete and continuous models, refer to the Derivation of Decisions from Classifiers section in chapter 3.

Individual Classifier Validation

The validation results of the M-IFN model of individual target variables are presented in Tables 6.1 and 6.2 for discrete and continuous variables respectively.

Work				
	Brier Score	**Sensitivity**	**Specificity**	**F-Measure**
Training set	0.2867	0.2852	0.8398	0.2088
Test set	0.2801	0.3015	0.8539	0.2213
More_Work_Ep				
	Brier Score	**Sensitivity**	**Specificity**	**F-Measure**
Training set	0.1135	0.1838	0.8703	0.1894
Test set	0.1395	0.1231	0.8546	0.1127

Table 6.1 Discrete target variables accuracy statistics

Model Name	Training set	Test set
Work_Dur	27%	26%
Ratio	28%	31%
Break_Time	78%	79%
Begin_Time	12%	12%

Table 6.2 Continuous target variable Relative Absolute Error (RAE)

It is shown that for the work target variable the M-IFN model reported rather a high accuracy in predicting the negative class (0) with a specificity of 0.84 for training set and 0.85 for the test set. The F-measure reported a weak predictive performance for the work target attribute. Furthermore, the Brier Score reported 0.29 and 0.28 for the training and test sets respectively. Considering the More_Work_Ep target attribute, the M-IFN reported higher accuracy in predicting the “0” class (one work episode) than the positive class, with a Specificity at 0.87 for the training set and 0.85. Similarly, looking at the F-Measure and the Brier Score it is noted the dropout of the values in test set compared to the training set. Considering the continuous target variables the accuracy is measured by calculating the Relative Absolute Error (RAE). RAE gives an indication of how good a predicted value is relative to the observed value. The M-IFN model reported high RAE for the Break_Time variable with 78 and 79 % for training and

test sets respectively. In addition, reported 27, 28, and 12 % for *Work_Dur, Ratio, Break_time* target variables respectively for the training set. And 26, 31, and 12 % for *Work_Dur, Ratio, and Break_time* target variables for the test set.

Activity Pattern Validation

The validation at the activity pattern sequence is reported in Table 6.3. As discussed in Chapter 4, the average length of the observed sequence for all activities is 5.02 symbols and 1.7 symbols for work activities. The average length of predicted all activity sequences is 3.3 symbols and 1.2 for work activities. To gain more understanding about the work activity sequences lengths, a confusion matrix to calculate the observed versus the predicted sequences lengths is generated. Table 6.3, shows the confusion matrix for work activity sequences lengths for training and test sets. The results reveal that the majority of work activity sequences contain one and two symbols. For sequences of length one, 81 and 83 percent for training and test sets respectively are correctly predicted. Additionally, the accuracy of predicting work activity sequences of length of four symbols is low at 10 and 4 percent for training and test sets respectively. Additionally, more than 60 percent of predicted work activities contain one symbol. While more than 30 percent of work activities contain two symbols and less than 10% of activities contain four symbols. The confusion matrix results show that the length of the work sequences are mainly of lengths one and two and this explains that the predicted work activities produced by M-IFN are short at an average of 1.2 symbols.

The SAM distance is not just influenced by the difference in symbols but also by the length of the sequence, i.e. the number of symbols (activities) in the activity sequence. Thus, to be able to interpret the validation at the activity pattern level, a confusion matrix of work activity sequence lengths is created.

Therefore, the SAM distances reported in Table 6.4 shows that the number of operations to equalize the predicted with the observed sequences is 4.74 and 4.7 for the training and test sets respectively. Therefore, given the weak performance

of the M-IFN method in predicting work activities, as shown by the individual classifier validation, results in shorter and diverse sequences for all activities. For example if in an observed activity sequence, a person go for shopping after having two work activity episodes (*H W B W S H*), but in the predicted sequence the sequence does not contain a work activity (*H S H*), this requires inserting three symbols to the predicted sequence which results in a higher SAM. For the work activity sequences, the length of predicted sequences is shorter than observed sequences. Thus, the SAM distances are 0.59 for the training set and 0.61 for the test set. Recall that the majority of the observed work activities (87%) contain one work episode, therefore, with an observed sequence length of 1.7 and predicted 1.2 symbols, a SAM distance of 0.6 is explainable.

	Training set	Predicted			
M-IFN	Sequence Length	1	2	4	Total
Observed	1	0.81	0.16	0.03	0.53
	2	0.38	0.58	0.04	0.36
	4	0.43	0.47	0.10	0.12
	Total	0.61	0.35	0.04	589.00
	Test set	Predicted			
M-IFN	Sequence Length	1	2	4	Total
Observed	1	0.83	0.16	0.01	0.52
	2	0.45	0.50	0.05	0.32
	4	0.42	0.54	0.04	0.16
	Total	0.64	0.33	0.03	298.00

Table 6.3 M-IFN work activity sequence lengths confusion matrices

SAM distance all activities pattern	
Dataset	M-IFN
Training	4.74
Test	4.7
SAM distance Work activity pattern	
Dataset	M-IFN
Training	0.592
Test	0.61

Table 6.4 M-IFN SAM distance for all and work activities

Work Activity Trip Matrix and Trips Start Time Level Accuracy Analysis (Spatial and Temporal Resolutions)

At the work activity trip matrix level (spatial resolution), the observed and predicted OD matrices were compared. An activity OD matrix contains the frequency of work activity trips for each combination of origins (rows) and destinations (columns). The frequency of trips at each zone in Flanders was aggregated forming a one dimensional array with work activity trip counts at each zone marginal, i.e. originating and arriving trips. The correlation is calculated between observed and predicted array entries ρ(observed, predicted). Table 6.5 shows that correlation coefficient for the work trips OD matrices are almost 0.83 for both the training and test sets. On the other hand, the work activity start time (temporal resolution) reveals a correlation coefficient of 0.84 and 0.82 between observed and predicted start times.

Work activity trip matrix level	
Dataset	**M-IFN**
Training	0.83
Test	0.83
Work activity start time per hour of the day	
Dataset	**M-IFN**
Training	0.84
Test	0.82

Table 6.5 M-IFN work activity trip matrix and work activity start time per hour of the day correlation coefficients

6.3 Upper bound: fully-informed set of classifiers

Comparisons of rule-based models and utility-maximizing models on activity travel patterns have been conducted (Arentze et al., 2000) and the rule-based system proved to be a very flexible approach. The rule-based system also performs well in predicting transport choice behaviour if an induction method is utilized (Wets et al., 2000 and Doherty, 2001). However, even if these induction models perform well at each decision step, they also show some limitations. The Performance of rule-based models using sequential scheduling process models

are affected by many factors. If the execution of a process model contains activation dependency, this results in prediction error propagation variation. Depending on the branching of decisions, the prediction error may lead to higher error rate for consequent decision steps. And thus, dramatically deprive model performance. On the other hand, the same prediction error may insignificantly worsen model performance. Another factor is the order of decision steps within the process model. There is no clear argument in the literature, except that the logical execution for the specific problem, why such decision steps order is used. However, and taking into account the activation dependency issue, there are decision steps that significantly influence model performance. Furthermore, interdependencies among decision steps in the process model may also influence model performance. The work activity process model in ALBATROSS is no exception.

Therefore, to solve the above mentioned issues, in the fully-informed approach the actual decision values are included as an input attributes in subsequent decision steps. Because we are using the actual values it is a non realistic scenario, but we consider it as an upper bound for the results. Continuous variables such as duration, Ratio, Inter (break time duration) and start time are discretized using Equal Frequency Interval (EFI) method, before being added. The datasets used in this approach are the same as shown in Table 4.1 in Chapter 4.

The analysis is performed on the work activity process model, which consists of six decision steps. Hence, to be able to analyze and assess the performance of the work activity process model only decision steps 1 and 3 are replaced by alternative classification methods. Because these decision steps branches the execution of the process model. Whereas the continuous decision steps (2, 4, 5 and 6) are kept unchanged using the original CHAID based tree induction. The analysis is conducted using three induction methods that are appropriate for analyzing the work activity process model. The first method is the original CHAID tree method. The second technique is the C4.5 decision tree method for two

reasons, (a) C4.5 is a benchmarking method in the data-mining community, (b) in a case study by Wets et al. (2000), and it has been found that the performance of CHAID and C4.5 decision tree algorithms are approximately equal in terms of goodness of fit. And this means that at the process model level, CHAID and C4.5 are expected to have approximately similar performance as well. The performance similarity between CHAID and C4.5 decision trees will support further analysis, especially when conducting experiments with the non-informed approach. Given that, CHAID is hard coded in ALBATROSS. And this is tailored with the inclusion of previous decision outcomes as input variables for subsequent decision steps. Thus, performing the analysis with new work process models using CHAID entails re-coding the entire module. The third technique is the Logistic Regression classification method, which will be referred to as Logit throughout this thesis. The Logit method was selected because it generally outperforms decision tree methods in terms of classification accuracy, especially for small size data sets, as shown by (Cox 1958). Moreover, Logit can produce probability estimates.

The C4.5 decision tree of the work (decision step 1) is illustrated in Figure 6.1. As mentioned in the previous section the rules generated by the M-IFN model represents the same rules of the decision tree up top the second level. The *wstat* variable at the root node of the tree and the *Day* attribute at the second level, where the model predicts a work (1) activity if it is a working day and no work activity if the day is a weekend.

The experiments were setup by running and deploying the model and generating schedules for both the training and test sets for 7 days and 5 times per day.
In the next subsections the model comparison criterion and validation results are discussed.

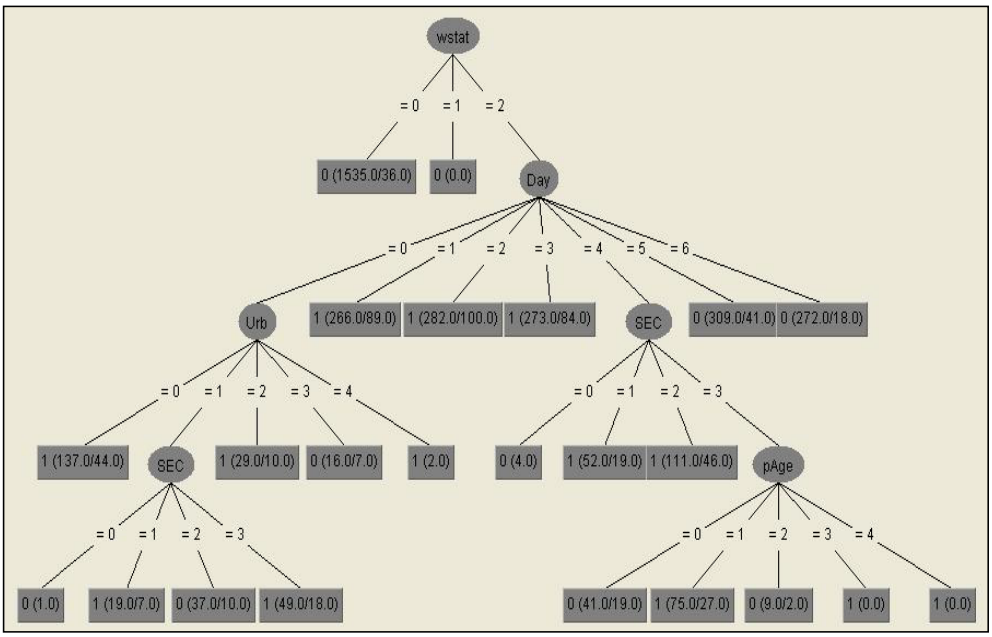


Figure 0.1 Work activity (decision step 1) C4.5 decision tree model

Classifier Level Accuracy Analysis

Table 6.6 provides the results of the analysis to assess model performance. The results suggest that for decision step 1, the Logit model outperforms all other methods specially in predicting the positive class value (*1*). As expected, CHAID and C4.5 show similar performance with a slight increase in performance in favor of C4.5, as illustrated by the BS and F-Measure. The predictive performance (sensitivity) for the (*1*) class variable, which is the minority class, is notably higher in the Logit approach. This can be explained by the fact that Logit is known to often outperform decision tree approaches for small size datasets (King and Zeng 2001). Results also show that the drop in the accuracy in the test set was not significant, while there was a slight increase in accuracy for the CHAID approach.

Considering the performance of decision step 3 (Table 6.7), again CHAID and C4.5 confirmed similar performance and outperform the Logit approach. The reason for the weaker performance of the Logit approach is that the data set at decision step 3 is highly skewed (87%). This leads to an underestimation of the rare class calculated by Equation 3.8 as reported by King and Zeng (2001).

Decision step - 1 Work	Training set			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
CHAID	0.11766	0.54065	0.841026	0.554455
Logit	0.113781	0.813008	0.839448	0.73026
C4.5	0.114957	0.59248	0.84497	0.594898
Decision step - 1 Work	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
CHAID	0.112366	0.554371	0.851653	0.563991
Logit	0.115959	0.791045	0.83628	0.704653
C4.5	0.115108	0.556503	0.825519	0.545455

Table 6.6 Fully-informed accuracy statistics for the Work model (classifier level)

Decision step - 3 More_Work_Ep	Training set			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
CHAID	0.106202	0.1875	0.890315	0.195122
Logit	0.0979974	0.125	0.985998	0.205128
C4.5	0.108202	0.242188	0.866978	0.227106
Decision step - 3 More_Work_Ep	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
CHAID	0.136115	0.155844	0.885496	0.179104
Logit	0.144773	0.0649351	0.959288	0.102041
C4.5	0.134237	0.194805	0.903308	0.230769

Table 0.7 Fully-informed accuracy statistics for More_Work_Ep (Nep) model (classifier level)

The continuous choice models which were trained using only the CHAID tree induction method used originally in ALBATROSS were kept the same for the analyses performed using alternative discrete choice models.

The performance of continuous choice models (Table 6.8) was assessed by means of the RAE. As depicted in Table 6.8 results showed fairly good results with 21, 18 and 8 % for decision steps 2, 4 and 6, respectively, for training sets,

and 20, 18 and 10 % for test sets, while for decision step 5 the RAE reported 64% for training and 61% for test set.

Decision Step	Name	Training set	Test set
2	**Work_Dur**	21%	20%
4	**Ratio**	18%	18%
5	**Break_Time**	64%	61%
6	**Begin_Time**	8%	10%

Table 6.8 Fully-informed RAE for continuous choice classifiers (decision steps)

Activity Pattern Level

Since experiments are conducted on the work process model, the SAM distance is calculated for both, all activities and for work activities in the schedule. The work activity patterns are expected to contain sequences of one, two or four symbols. It contains one symbol when no work activity is predicted (e.g. H). Furthermore, the sequence contains two symbols when the schedule includes one work episode (e.g. H W). On the other hand, work activity patterns contain sequences of four symbols when the schedule contains two work episodes with a break time in between (e.g. H W B W).

The average length of observed activity patterns is 5.025 (2.095) with standard deviation between brackets, whereas activity sequences average lengths for CHAID, C4.5, and Logit models are 3.61 (1.73), 3.92 (1.9), and 3.74 (1.51) respectively. For the work activity sequences, the average length of observed sequences is 1.7. While for CHAID, C4.5 and Logit models the average lengths are 1.45 (0.72), 1.5 (0.75), and 1.32 (0.8) respectively. The confusion matrices, for training and test sets, between observed and predicted activity pattern lengths for all models are reported in Tables 6.9, 6.10, and 6.11. The results show that CHAID and C4.5 reported comparable results. For sequences of lengths one and four symbols, decision tree models outperformed the Logit model however, for sequences of lengths two, Logit model reported higher accuracy. This can be explained by the fact that the Logit model outperformed decision tree models in decision step 1 (Work model), which means more accurate work activities are

predicted. In addition, for decision step 3 (More_Work_Ep), the Logit model reported weaker performance than decision tree models. This means that the majority of the work activity sequences contain two work episodes (e.g. W H). And therefore, the work activity sequence contains two symbols.

The confusion matrix results also explain that decision tree models reported higher work activity sequences lengths.

	Training set	Predicted			
CHAID	Sequence Length	1	2	4	Total
Observed	1	0.81	0.16	0.03	0.52
	2	0.42	0.54	0.04	0.36
	4	0.40	0.50	0.10	0.12
	Total	0.64	0.32	0.05	589.00
	Test set	Predicted			
CHAID	Sequence Length	1	2	4	Total
Observed	1	0.83	0.14	0.03	0.52
	2	0.41	0.56	0.03	0.32
	4	0.50	0.44	0.06	0.16
	Total	0.64	0.33	0.03	298.00

Table 0.9 Fully-informed CHAID model work activity sequence lengths confusion matrices

	Training set	Predicted			
C4.5	Sequence Length	1	2	4	Total
Observed	1	0.81	0.15	0.04	0.52
	2	0.37	0.55	0.07	0.36
	4	0.45	0.45	0.10	0.12
	Total	0.61	0.33	0.05	589.00
	Test set	Predicted			
C4.5	Sequence Length	1	2	4	Total
Observed	1	0.82	0.16	0.02	0.52
	2	0.40	0.51	0.09	0.32
	4	0.34	0.53	0.13	0.16
	Total	0.61	0.33	0.06	298.00

Table 6.10 Fully-informed C4.5 model work activity sequence lengths confusion matrices

Logit	Training set	Predicted			
	Sequence Length	1	2	4	Total
Observed	1	0.75	0.22	0.03	0.52
	2	0.10	0.87	0.03	0.36
	4	0.10	0.83	0.07	0.12
	Total	0.44	0.54	0.02	589.00
Logit	Test set	Predicted			
	Sequence Length	1	2	4	Total
Observed	1	0.72	0.28	0.00	0.52
	2	0.09	0.90	0.01	0.32
	4	0.10	0.85	0.05	0.16
	Total	0.42	0.57	0.01	298.00

Table 0.11 Fully-informed Logit model work activity sequence lengths confusion matrices

The SAM distances, for training and test sets, between observed and predicted activity pattern for all models are reported in Table 6.12. As expected CHAID reported approximately similar SAM distance with C4.5, where Logit reported higher SAM distance than C4.5 and CHAID. By means of SAM distance the Logit model reported higher distance than decision tree models. And the C4.5 decision tree model predicted longer activity patterns than all other models and thus, reported lower SAM distance.

SAM distance all activities pattern			
	CHAID	Logit	C4.5
Training	4.18	4.511	4.12
Test	3.87	4.351	3.94

Table 6.12 Fully-informed All activities SAM distance

Table 6.13 shows the SAM distance for work activities sequences. Again decision tree models (CHAID and C4.5) reported comparable results, with a slightly lower SAM in favour of CHAID. The Logit model reported higher SAM than both CHAID and C4.5. As shown in the confusion matrices of the Logit model (Table 6.11) the majority of the work sequences are predicted with length of two symbols. This indicates that more operations (deletion, insertion, or substitution) are required to equalize the predicted to observed sequences

SAM distance work activity pattern			
	CHAID	Logit	C4.5
Training	0.497524	0.590391	0.529965
Test	0.480931	0.593363	0.521545

Table 6.13 Fully-informed Work activities SAM distance

Work Activity Trip Matrix and Trips Start Time Level Accuracy Analysis (Spatial and Temporal Resolutions)

In Table 6.14 the training and test sets correlation coefficient between observed and predicted work trips OD matrices for all models is reported. The correlation coefficients for decision tree models (CHAID and C4.5) are similar, while the Logit model reported quite lower correlation.

	Work activity trip matrix level		
Dataset	CHAID	Logit	C4.5
Training	0.832	0.81	0.84
Test	0.816	0.8	0.82

Table 6.14 Fully-informed Work activity trip matrix correlation coefficients

The work activity start time level (temporal resolution) was also evaluated by calculating the correlation between the observed and predicted work activity start times for each hour of the day. The results in Table 6.15 indicate that the correlation coefficients are similar with the Logit approach having a slightly lower correlation coefficient than the CHAID and C4.5 approaches.

	Work activity start time per hour of the day		
Dataset	CHAID	Logit	C4.5
Training	0.896	0.873	0.9
Test	0.827	0.771	0.85

Table 6.15 Fully-informed Work activity start time per hour of the day correlation coefficients

6.4 Non-informed set of classifiers

In the fully-informed approach actual values of decision outcomes are included as an input attributes in subsequent decision steps. However, the non-informed approach suggests that models, at individual decision steps, are trained without

the inclusion of previous decision outcomes as attributes in the next decision step. The outcomes of previous decision models are already included in the data sets of successive decision models. However, in this approach these decision outcomes are eliminated, and the same set of attributes are used to train the models at decision steps. The datasets used to train all models in the work activity process model for the non-informed approach is shown in Table 4.1. As explained in chapter 4, the datasets contain situational and socio-demographic variables that are used as prediction variables in FEATALB.

In the FEATALB framework, the CHAID decision trees are trained and constructed as part of the system. And thus, the training attribute lists and names are hard coded within the model. For this reason, the training and deployment of the CHAID decision trees requires re-coding of the whole approach. Hence, the *DecisionMaker PMML* implementation in the system can be used. And considering the fact that decision tree models obtain similar performance (as shown in the results of the analysis in the previous section), the CHAID decision trees can be replaced by other decision tree models.

In this approach, discrete choice models, i.e. decision steps 1 and 3 are experimented using C4.5 and Logit models. Nevertheless, for continuous models the CART decision trees is used. In ALBATROSS the derivation of decision rules employs a probability distribution among classes for discrete models. For continuous decision trees utilized in FEATALB, the distributions at each leaf node specifying *m-1* cutoff points and the minimum and maximum of the range. For this reason the simulation is run 10 times for each day of the week seven days a week.

The results of the non-informed approach are validated at three levels. First, at each decision steps level, second, on the activity pattern level and thirdly, at the spatial and temporal network level. This validation is performed to confirm that the model is applicable on the study area from which the data was collected.

Classifier Level Accuracy Analysis

As discussed in chapter 3, the evaluation criteria for discrete choice models are presented using two accuracy measures, the confusion matrix accuracy measures. And the Brier score (Brier 1950) because of the probabilistic action assignment rule used in scoring the models.

The results for discrete choice decision steps are reported in Tables 6.16, and 6.17. Table 6.16 shows accuracy statistics for decision step 1 (include work). The results illustrates that the Logit model outperforms the C4.5 decision tree model. The sensitivity measure for the Logit model indicates that its performance in predicting the positive class (1) is high. Such good performance can be also detected in the F-measure and the Brier score. The predictive performance of the logit model can be explained because of the fact that logit is known to perform well for small size data sets as reported by King and Zeng (2001).

Decision step - 1 Work	**Training set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.113781	0.813008	0.839448	0.73026
C4.5	0.114957	0.59248	0.84497	0.594898
Decision step - 1 Work	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.115959	0.791045	0.83628	0.704653
C4.5	0.115108	0.556503	0.825519	0.545455

Table 6.16 Non-informed approach accuracy statistics for include work activity (decision step 1)

For decision step 3 (*More_Work_Ep*) accuracy statistics for the non-informed approach, indicates that the C4.5 decision tree model outperforms the Logit model as shown in Table 6.17. The reason for the weak performance of the Logit approach is that the data set for the number of wok episodes is highly skewed (87%). This leads to an underestimation of the rare class calculated by Equation 3.8 as reported by King and Zeng (2001).

It is also noted that the models in fully-informed approach outperforms the models in the non-informed approach. This implies that on the single classifier level, the attributes interdependencies have an added-value.

The continuous choice models which were trained using the CART decision tree were kept the same for C4.5 and Logit models. However, in the full approach, the CART trees were trained using actual values from previous decision outcomes i.e. using the attribute dependency feature. On the other hand, in the independent approach, the CART trees were trained without inclusion of decision outcomes from previous decision steps.

Decision step - 3 More_Work_Ep	Training set			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.113867	0.0625	0.994166	0.111475
C4.5	0.11308	0.109375	0.873979	0.112
Decision step - 3 More_Work_Ep	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.146908	0.012987	0.982188	0.0235294
C4.5	0.138436	0.142857	0.882952	0.164179

Table 6.17 Non-informed accuracy statistics for Number of work episodes (decision step 3)

The performance for continuous choice models were evaluated by means of the Relative Absolute Error (RAE). The reason for choosing this measure is that it is provided as percent error measure for numeric predictions. The results are shown in Tables 6.18. Results show that RAE for decision steps 2, 4 and 6 (work duration, Ratio, and work start time) are 21, 22, and 9 % respectively for training sets, and 20, 22 and 11 % for test sets. While for decision step 5 (break time) reported 65 and 67 % for training and test sets respectively. Considering the results of continuous models for the fully-informed approach, the RAE is lower than the models in the non-informed approach. Except for the *Work_Dur* decision model, because no decision outcomes are added to this model.

Decision Step	Name	Training set	Test set
2	**Work_Dur**	21 %	20%
4	**Ratio**	22 %	22 %
5	**Break_Time**	65 %	67 %
6	**Begin_Time**	9 %	11 %

Table 6.18 Non-informed approach RAE for CART continuous choice classifiers

Activity Pattern Level

For the non-informed approach, the SAM distance between observed and predicted activity sequences are calculated for all activities in the schedule and for work activities. As discussed in Chapter 4, the average length of observed activity patterns is 5.025, whereas work activity average lengths for C4.5, and Logit models are 3.50, 3.35 respectively. The C4.5 decision tree model predicted longer activity patterns than the Logit model. The observed work activity pattern average length is 1.7. While for the C4.5 and Logit work activity pattern lengths reported 1.45 and 1.3 respectively. The confusion matrices for work activity sequences lengths for both models are shown in Tables 6.19 and 6.20. The results show that the accuracies of predicting the correct lengths are lower than the fully-informed approach for all sequences lengths. For the training and test sets, the C4.5 model's accuracy in correctly predicting work activity sequences lengths are lower than the fully-informed approach. The Logit model reported comparable results with those of the fully-informed approach. However, it is poor in predicting work activities with four symbols.

As shown in Table 6.21, the C4.5 model obtained lower SAM distances than Logit models at both the all activities and the work activities sequences. Considering the fully-informed approach, the length of all and work activities is higher than the length of the work activities in the non-informed approach, which closer to the length of the observed lengths. It is also noted that on the activity pattern level the fully-informed approach performed better than the non-informed approach.

Work Activity Trip Matrix and Trips Start Time Level Accuracy Analysis (Spatial and Temporal Resolutions)

At the work activity trip matrix level (spatial resolution), the observed and predicted OD matrices are compared using a correlation coefficient. Similarly the observed and the predicted work activity start times (temporal resolution) are

compared via a correlation coefficient. The correlation coefficient measures the relation between observed and predicted values.

	Training set	Predicted			
C4.5	Sequence Length	1	2	4	Total
Observed	1	0.79	0.20	0.02	0.52
	2	0.39	0.54	0.07	0.36
	4	0.44	0.47	0.09	0.12
	Total	0.60	0.35	0.04	589.00
	Test set	Predicted			
C4.5	Sequence Length	1	2	4	Total
Observed	1	0.76	0.21	0.03	0.52
	2	0.35	0.56	0.08	0.32
	4	0.50	0.42	0.08	0.16
	Total	0.59	0.36	0.05	298.00

Table 0.19 Non-informed C4.5 model work activity sequence lengths confusion matrices

	Training set	Predicted			
Logit	Sequence Length	1	2	4	Total
Observed	1	0.73	0.26	0.00	0.52
	2	0.12	0.88	0.00	0.36
	4	0.10	0.88	0.01	0.12
	Total	0.44	0.56	0.00	589.00
	Test set	Predicted			
Logit	Sequence Length	1	2	4	Total
Observed	1	0.72	0.28	0.00	0.52
	2	0.09	0.90	0.01	0.32
	4	0.08	0.92	0.00	0.16
	Total	0.42	0.58	0.00	298.00

Table 0.20 Non-informed Logit model work activity sequence lengths confusion matrices

	SAM distance all activities		SAM distance work activities	
	Logit	C4.5	Logit	C4.5
Training	4.68	4.37	0.63	0.57
Test	4.66	4.51	0.62	0.55

Table 0.21 Non-informed all activities and work activity sequences SAM distances

As mentioned in chapter 4, work trip frequencies at each zone in Flanders are aggregated to form a one dimensional vector. This is done to eliminate low trip

counts at each cell. Table 6.22 illustrates the correlation coefficients for work activity trip matrices and work activity start times. Results show that the C4.5 decision tree models reported higher correlation than the Logit models for the training and test sets.

It is also noted that in the non-informed approach obtained less correlation coefficients than the fully-informed approach.

	Work activity trip matrix level	
Dataset	**Logit**	**C4.5**
Training	0.8	0.84
Test	0.79	0.82
	Work activity start time per hour of the day	
Dataset	**Logit**	**C4.5**
Training	0.86	0.88
Test	0.78	0.85

Table 0.22 Non-informed approach work activity trip matrix and start time per hour of the day correlation coefficients

6.5 Discussions

In this chapter the work activity process model in the FEATALB framework is experimented by running the model in three different settings:

1- Modeling the decision steps in the work activity process model using a multi-target classification method (M-IFN), where all models are predicted simultaneously.

2- Fully-informed approach, where three classification methods (CHAID, C4.5, and Logit) are used in discrete choice decision steps (Work, More_Work_Ep) and CHAID decision trees for continuous models. The models are trained preserving the attributes interdependencies among decision models. Further while the simulation is running the actual decision outcomes are included as attributes to subsequent decision steps rather than using the predicted decision outcomes.

3- Non-informed approach, using C4.5 and Logit models for discrete choice models and CART decision trees for continuous models. In this approach the models are trained while eliminating the attributes interdependencies.

Base Line

Both M-IFN and the non-informed approaches are designed such that the attributes interdependencies feature is eliminated. Therefore, they are process independent and can be considered as lower bound models. The validation results for the test set of decision step 1 (Table 6.23) shows that the M-IFN is inferior, in terms of performance, to the non-informed approach. The C4.5 model was represented because its performance is weaker than Logit. On the other hand, for decision step 3 the validation results are represented in Table 6.24. Due to the fact that the Logit technique fails for decision step 3, we choose the C4.5 model as technique in the non-informed approach. The results show that the C4.5 model in the non-informed approach outperformed the M-IFN approach.

Approach	1-Brier Score	Sensitivity	Specificity	F-Measure
M-IFN	0.72	0.30	0.85	0.22
Non informed – C4.5	0.88	0.56	0.83	0.56

Table 0.23 Decision step 1 (Work) model accuracy statistics

Model	1-Brier Score	Sensitivity	Specificity	F-Measure
M-IFN	0.86	0.12	0.85	0.11
C4.5	0.86	0.14	0.88	0.16

Table 6.24 Decision step 3 (More_Work_Ep) model accuracy statistics

Table 6.25 shows the RAE for continuous models of both approaches (M-IFN and non-informed). The M-IFN reported higher RAE than the decision tree models in continuous models.

Decision Step	Name	M-IFN	Non-informed
2	**Work_Dur**	26 %	20%
4	**Ratio**	31 %	22 %
5	**Break_Time**	79 %	67 %
6	**Begin_Time**	12 %	11 %

Table 6.25 RAE for continuous models

The results at the activity pattern level (Table 6.26), the observed mean activity all activities length is 5.02 symbols and 1.7 for work activities. The M-IFN approach obtained smaller activities than the

C4.5 model in the non-informed approach. Therefore, reported a higher SAM distance.

	Observed	M-IFN	Non-informed
All activities	5.02	3.3	3.5 (C4.5)
Work Activities	1.7	1.2	1.45 (C4.5)

Table 6.26 Mean length of all and work activities for M-IFN and on-informed approaches

At the spatial and temporal dimensions the validation results (Table 6.27) shows that at the work activity both approaches reported comparable results. However, at the work activity start time, the non-informed approach obtained higher correlation coefficient than the M-IFN approach.

	M-IFN	Non-informed
OD Work activity trips	0.83	0.82
Work activity start time	0.82	0.85

Table 6.27 Spatial and temporal correlation coefficients

Based on the discussion above, we can conclude that there is a clear difference between both approaches, specifically for the sensitivity criterion for discrete decision steps and for continuous decision steps. Moreover, the validations at all levels suggest that the non-informed approach outperformed M-IFN. The results of the M-IFN approach indicate that this approach serves as the base line model. Hence, the results of this approach will be used as the base line figures.

Target value and Range

In the fully-informed approach, the actual values are included as attributes to subsequent decision steps. Therefore, it is expected to give us the target performance values. We have however to remember that the fully-informed approach is process model dependent.

As target values for the work model (decision step 1) we take the maximum accuracies of table 6.6 i.e. the C4.5 method (Table 6.28). For decision step 3 the

Logit model again we exclude the Logit model (Table 6.29). In addition, target and base line values (RAE) for continuous models are shown in Table 6.30. The results show that the fully-informed approach provides a performance improvement range for them models. Therefore, comparing baseline and target values shows the different ranges and the possibilities to improve the model. Table 6.31 illustrates the results at the SAM validation level, and it is obvious that the fully-informed approach predicted longer all and work activity sequences. Thus, reported lower SAM distance. The figures at this level suggest that the M-IFN approach is considered as the lower bound.

Approach	1-Brier Score	Sensitivity	Specificity	F-Measure
Base line	0.72	0.30	0.85	0.22
Target	0.88	0.81	0.84	0.73

Table 0.28 Baseline and target decision step 1 (Work) model accuracy statistics

Model	1-Brier Score	Sensitivity	Specificity	F-Measure
Base line	0.86	0.12	0.85	0.11
Target	0.87	0.19	0.90	0.23

Table 0.29 Baseline and target decision step 3 (More_Work_Ep) model accuracy statistics

Decision Step	Name	Baseline	Target
2	**Work_Dur**	26 %	20%
4	**Ratio**	31 %	20 %
5	**Break_Time**	79 %	61 %
6	**Begin_Time**	12 %	10 %

Table 6.30 Baseline and target RAE for continuous models

	Baseline	Target
All activities	3.3	3.94 (C4.5)
Work Activities	1.2	1.5 (C4.5)

Table 0.31 Baseline and target Mean activities (All and Work) sequences lengths

Table 6.32 shows the baseline and target value at the spatial and temporal aggregated level. The results indicate that the range between the baseline and target values is too small to be useful as a performance boundary measure. This

concludes that the validation at the spatial and temporal levels is not sensitive to local changes of the process model.

	Baseline	Target
OD Work activity trips	0.83	0.82
Work activity start time	0.82	0.85

Table 0.32 Baseline and target spatial and temporal correlation coefficients

Attributes interdependencies (process model dependent) feature added value

The analyses in this chapter suggest that the generated M-IFN model is a set of rules that are similar to those generated by the decision tree models (specifically the decision tree for decision step 1 - Work) up to the second level (Figure 6.1). In addition, the M-IFN classification method has the lowest fitting capacity. Hence, the M-IFN model can be thought of as the baseline performance (Lower bound) model. Furthermore, the models in the Fully-informed approach, which is a process model dependent, outperformed models in both M-IFN and non-informed approaches. Similarly in terms of performance boundaries, the fully-informed approach can be seen as the upper performance boundary. While training the models in the non-informed approach, it is noted that the performance dropped. However, the models in this approach still outperformed the M-IFN model.

Regarding the performance of the M-IFN model compared to the individual classifier approaches conform to results obtained in the literature. As presented by Caruana (1993) and Piccart (2008), that the combination of several single-target classifiers may increase the overall predictive performance.

The validation results of the fully-informed and non-informed approaches, illustrated that attributes interdependencies among models in the work activity process model in ALBATROSS increase the predictive performance. At the individual classifier level running the fully-informed approach improves the quality of the predicted schedules, although it is a non-realistic scenario. This is also reflected at the activity pattern level as the full approach proved to have more similar patterns than the non-informed approach. However, the validation at the spatial and temporal resolutions was not sensitive to capture local changes in the

process model. Thus, to be able explain the effect of including decision outcomes, which are used as attributes in subsequent decision steps, the Relief-F feature selection technique is used. The Relief-F technique reports the most relevant attributes in descending order that are important in predicting the class variable.

Table 6.33 depicts the result of running the Relief method implemented in WEKA (Hall et al., 2009). The *Work_Dur* attribute, which is the duration of the work activity, and the decision outcome of decision step 2, is the first ranked attribute that is relevant in identifying the *More_Work_Ep* class (decision step 3), and the Ratio (decision step 4).

Decision step	Name	Relief weight	Ranked attributes
3	Number of work episodes (More_Work_Ep)	0.12439	Work_Dur
		0.09553	Xdag
		0.09106	Xarb
		0.09035	Ddag
		0.08801	Xn_dag
4	Ratio	0.040334	Work_Dur
		0.016735	Urb
		0.015106	Xn_dag
		0.013993	wstat
		0.008867	Comp
5	Break Time	0.032438	pAge
		0.030761	Day
		0.027669	Ncar
		0.025479	Work_Dur
		0.023602	Ddag
6	Start time	0.0238621	Day
		0.0128535	More_Work_Ep
		0.0112649	wstat
		0.0088725	Driver
		0.0064544	Work_Dur
		0.0000178	Break_time

Table 6.33 Relief feature selection results for datasets used in decision steps in the work activity process model.

In addition the *Work_Dur* attribute ranks the fourth in the list of important attributes to correlate to the break time variable (decision steps 4). For Start time variable (decision step 6), again the *Work_Dur* attribute, the *More_Work_Ep*

(decision outcome of decision step 3) and the *Break_time* (decision outcome of decision step 5) are among the list of the first six relevant attributes.

The results of the Relief feature selection analysis verify that including decision outcomes of decision steps as attributes in successive decision will improve the performance of models in rule-based activity based models.

Finally in this chapter, performance boundaries have been identified. These performance boundaries can be used to asses the performance of other models or other decision sequences, as will be presented in the next chapter. To illustrate this, let us consider the individual classifier level validation results for the three approaches (M-IFN, fully-informed, and non-informed). Figure 6.2 depicts the Brier Score (BS) of the three approaches for discrete choice models, recall that a lower BS values indicates a better model. The similar BS value of the work model (decision step 1) in the fully-informed and non-informed approaches is because it is the same model used in both approaches as it is the first step in the process model.

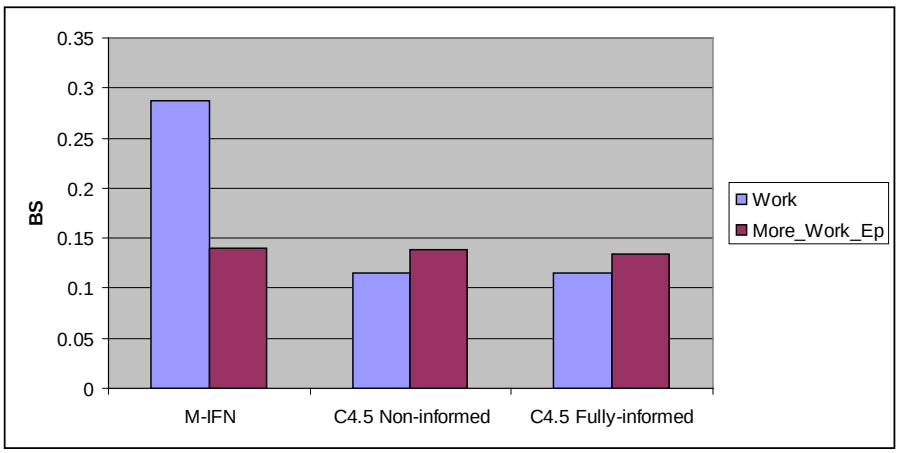


Figure 0.2 Brier Score of the M-IFN, Fully-informed, and Non-informed approaches

As for the continuous models, Figure 6.3 shows the RAE obtained by the three approaches. The RAE of continuous models obtained by the M-IFN model reported higher error, while the fully-informed approach reported the lowest error

rates. The Work_Dur model reported similar RAE value since no decision outcome is added to this decision step.

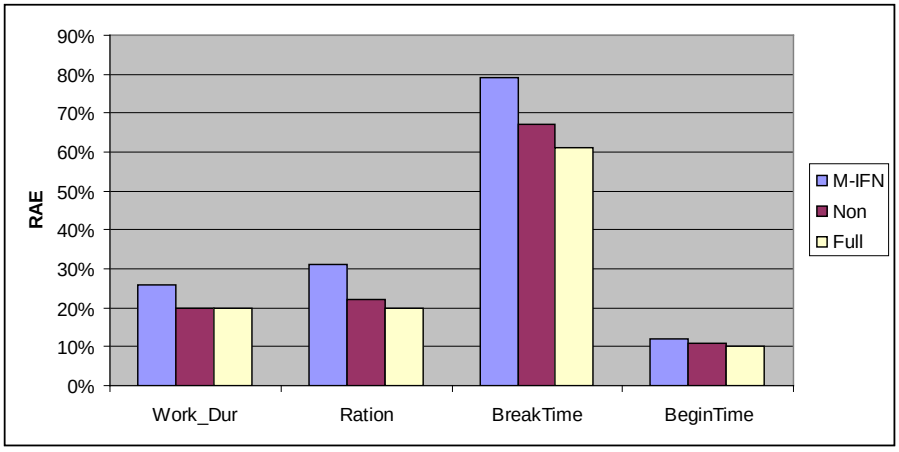


Figure 6.3 RAE for the M-IFN, Fully-informed, and Non-informed approaches

At the activity pattern level, the SAM distances for both all activities and work activities for the three approaches conforms also to the classifier level accuracy results. Figure 6.5 illustrates the comparison of SAM distances having the M-IFN with the highest SAM distances and the fully-informed approach with the lowest SAM distances.

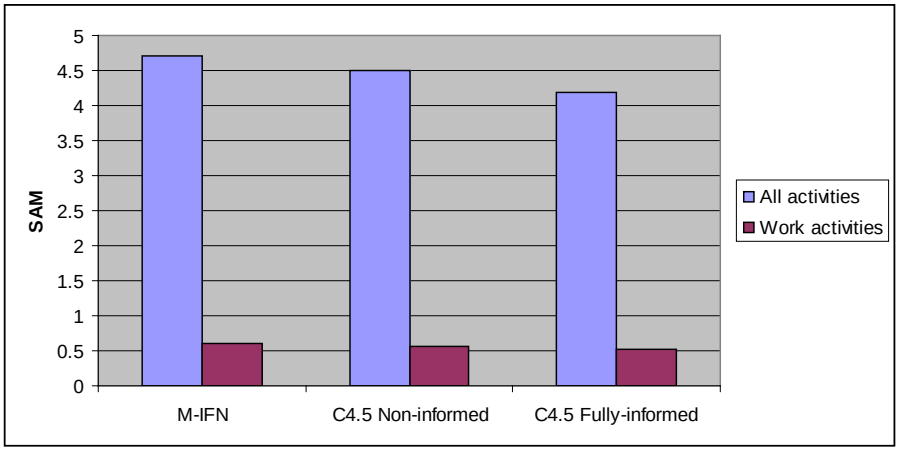


Figure 6.4 SAM distances for the M-IFN, Fully-informed, and Non-informed approaches

6.6 Conclusions

Process models in rule-based activity-based models contain decision steps where each decision step utilizes a classification model to predict relevant attributes. The work activity process model in ALBATROSS consists of six decision steps. At each decision step a CHAID based decision tree model is used. The analyses in this chapter aimed at identifying performance bounds for process models. The six decision steps in the process model was trained and deployed in three settings or approaches. Firstly, the decision steps are modeled in one model using a multi-target classification method (M-IFN). Secondly, using the fully-informed approach where decision outcomes are added as attributes in the data sets of consequent decision steps. Thirdly, with the non-informed approach where the models at decision steps are trained without the inclusion of decision outcomes i.e. without the attributes interdependencies between decision steps. The validation results showed that, as suggested in the literature, several single-target classification methods outperform multi-classification methods. In addition, decision tree models obtained comparable performance. The results also revealed that the model achieved by the M-IFN is similar to that obtained by the decision tree model up to the second level. Therefore, the M-IFN model can be used as a lower performance bound (base line model) for all other models. On the other hand, using the fully-informed approach, where the observed values are added as attributes instead of the predicted outcomes, outperformed the M-IFN and the non-informed approach can be thought of as the upper performance abound. Since, no other models are expected to reach this performance. Finally, the results suggest that the attributes interdependencies among models in the decision steps increase the performance. This is further confirmed by investigating the datasets using a feature selection method. As the results verified that the decision outcomes as attributes in the data sets for consecutive

decision steps are listed among the most relevant attributes in identifying the class variable.

References

Arentze, T., Hofman, F., van Mourik, H., Timmermans, H. and Wets, G. (2000) Using decision tree induction systems for modeling space-time behavior. Geographical Analysis, 32, 330-350.

Beau Piccart, Jan Struyf, and Hendrik Blockeel. (2008). Empirical Asymmetric Selective Transfer in Multi-objective Decision Trees. In Proceedings of the 11th International Conference on Discovery Science Pages 64 – 75.

Caruana, R. (1997) Multitask Learning. Machine Learning, 28, pp. 41–75.

Caruana, R. (1993). Multitask Learning: A Knowledge-Based Source of Inductive Bias. Proceedings of the 10th International Conference on Machine Learning, ML-93, University of Massachusetts, Amherst, pp. 41-48.

D. R. Cox. (1958) The regression analysis of binary sequences. Journal of the Royal Statistical Society, Series B (Methodological), 20(2): pp 215–242.

Doherty, S. (2001) Classifying activities by time horizon using machine learning algorithms. Paper presented at the 80th Annual meeting of the Transportation Research Board, Washington, D.C., USA.

Hall M. A., Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

King, G. and Zeng, L. (2001). Logistic regression in rare events data. Political Analysis, 9:137-163.

M. Last, Multi-objective Classification with Info-Fuzzy Networks. ; In Proceedings of ECML. 2004, 239-249.

E. Suzuki, M. Gotoh, and Y. Choki (2001). Bloomy Decision Tree for Multi-objective Classification. L. De Raedt and A. Siebes (Eds.): PKDD 2001, LNAI 2168, pp.436 –447.

Wets G, Vanhoof K, Arentze T A, Timmermans H J P. (2000) Identifying decision structures underlying activity patterns: an exploration of data mining algorithms" Transportation Research Record number 1718, 1 – 9.

Chapter 7

Experimenting Process Model Sequences (Data Representation) for Work activity process Models

In ALBATROSS, which is a rule-based activity based model, a sequential scheduling process model is assumed to generate a schedule. However, the current process model is based on the researcher's experience in activity-based models. The aim of this chapter is to study the effect of rearranging decision steps on the performance of rule-based activity-based models. In other words, finding a better data representation introduced in process models. This goal is achieved by introducing new scheduling process models. Two alternative work activity scheduling process models, other than the original process model, are developed, integrated, and utilized in FEATHERS. Furthermore, for discrete choice models in each process model, two classification methods are used, namely C4.5 decision trees and logistic regression. While CART decision trees are trained and deployed for continuous models. The process models are executed for simulation using the Flanders data. Consequently, the performance of each process model is validated at three levels, as presented in previous chapters: Firstly, at each decision steps level, secondly on the activity pattern level and thirdly, at the spatial and temporal network level. Introducing other work activity process models allows for gaining more understanding of the scheduling engine in ALBATROSS. And further elucidate the impact of changing the order of decision steps (activation dependency) within the process models on the predictive performance of the model.

The chapter begins by the introduction which includes an explanation about the scheduling engine of rule-based activity-based models, specifically ALBATROSS. Alternative work activity process models are introduced and explained in details. In addition, experiments design and induction methods used to analyse all proposed process models are elaborated. Then the validation

results are presented and discussed. Finally the chapter ends with the discussion and conclusions.

7.1 Introduction

Computational process models constitute a powerful theoretical approach that conceptualizes choices as outcomes of using context-dependent heuristics. ALBATROSS, as a rule-based computational process activity-based model, consists of a series of agents that together handle the consistency of the data. The core of the ALBATROSS framework is the scheduling engine which controls the scheduling processes as a sequence of decision steps. At each decision step the scheduling engine classifies the condition information for making a key decision. Hence, computational process rule-based models are based on a set of models (26 models in ALBATROSS) that represent transport choice behaviour. Alternatively, defining interdependence among decision outcomes within the scheduling engine depends on the ordering of decision steps. As a result, such models generate a feeling of a black box (Timmermans et. al., 2002). One of the aims of the analysis in this chapter is to eliminate the black box complication of the system. This is achieved by conducting experiments on the work activity process model in ALBATROSS. In addition to the originally used process model, two work activity process models are developed, and plugged-in to the scheduling engine in the framework. Moreover, for each process model, two induction methods are utilized at each decision step. This allows for gaining more understanding about the process model. And whether changing the order of decision steps in the work process significantly affects the predictive performance of the model.

In the next section, the process model sequences are explained in details. In addition, the rationale behind finding better data representation is discussed. Followed by the illustration of the three process models developed in the context of alternative data representation.

7.2 Work Activity Process Model Sequences

As discussed in previous chapters, the work activity process model in ALBATROSS is the first component in the scheduling process model. This makes it an important component, because, based on the decision outcome of this process model it is decided that the person's schedule will contain a work activity or not. Furthermore, if the schedule contains a work activity, other schedule related attributes will be decided, such as, the duration of the work activity, break time duration (if two work episodes are predicted), and the start time of the work activity. Thus, an empty schedule's time scale is filled with an activity type and its related information, which will influence the duration of other activities further as the execution of the scheduling process proceeds. Other issues that influence the predictive performance of the process model are the **activation dependency** feature and the **attributes interdependencies** between decision models in the work activity process model.

As discussed in chapter 6, the including of previous decision outcomes as attributes to subsequent decision steps enhances the predictive performance of the model, under the condition that such attributes are indicated as relevant in predicting the class attribute. Therefore, changing the order of the decision steps is expected to influence the predictive performance of the model. To this end, and to achieve a better understanding of the work activity process model, two process models, other than the original, are developed by changing the order of decision steps. The occurrence of the activation dependency feature is shifted either up or down in the process model. Additionally, the attribute dependency feature will vary depending on the disposition of decision steps in each process model. The original work activity process model and the proposed alternative process models are shown in Figures 7.1, 7.2 and 7.3 respectively.

For ease of reference, the original work activity process model will be referred to as process model 1. While the alternative process models will be referred to as process model 2 and process model 3.

For each process model, the simulation is run using two different induction methods for discrete choice decision steps. The first induction method is C4.5

decision tree models, and the second using Logit models. The continuous choice decision steps are trained and utilized using Classification and Regression Trees (CART) models. The C4.5 decision tree and the logit models were trained and generated using the Rattle package for R (Williams 2009). The models were then exported to PMML using the *decisionMaker* class implemented in the FEATHERS framework to deploy PMML decision trees and Logistic regression. To avoid over fitting, the minimum number of cases is set to 30 in the C4.5 decision tree models. This number was also set in the original CHAID decision trees used in the ALBATROSS model developed by Arentze and Timmermans (2005).

Conceptually the number alternative process models for a process model with six decision steps can be $6! = 640$ process models (data representation). However, logical process models that can be adapted to predict work activities in the sense of rule-based activity-based models are few, since the order of decision steps (type of model) to predict is important. For example, one cannot predict the duration of a work activity or number of work episodes if the no work activity has been predicted. Similarly, the ratio between work episodes and the break time duration could not be specified if only one work episode is predicted. Nevertheless, other decision steps can be reordered, e.g. the number of work episodes or the work activity start time can be predicted before the work duration. This rationale will be used in determining alternative process models. Hence, for any process model to make sense, the work activity model (decision step 1) must be consulted first.

The stochastic approach to derive decision rules from decision trees and Logit models, as discussed in Chapter 3, is also used in this chapter. The models for each process model are validated at three levels: (1) The individual classifier level, using confusion matrix accuracy statistics, (2) the Activity pattern level using the SAM distance measure, and (3) the spatial-temporal resolution, by calculating the correlation coefficient between observed and predicted work trips at each zone, and work activity start times. The validation results of The C4.5 decision trees and Logit models for Process model (1). As illustrated by figures

7.1, 7.2, and 7.3, decision step 1 (include work activity choice) for all process models is the same and hence, at the individual classifier level the accuracy and validation results are the same for both C4.5 and Logit models. Finally, in process model 3, the decision step 2 (number of work episodes) contains no attribute inclusion from previous decision step and so it is the same as the non-informed approach model analysed in chapter 6.

7.2.1 Process Model 1

Figure 7.1 depicts process model 1 (the original work activity process model) adapted in ALBATROSS. It is shown that the activation dependency feature in process model 1 occurs at decision steps 1 and 3.

The process model begins by evaluating the work activity model according to the classifier utilized in this decision step. If no work (0) activity is predicted, then the process terminates. If a work activity (1) is predicted, then the work activity duration is predicted. Followed by decision step 3 (More_Work_Ep) which is responsible for determining the number of work episodes (0: one work episode, 1: two work episodes). If one work episode is predicted then the execution proceeds to decision step 3 (work activity start time) hereafter the process terminates. On the other hand, if two work episodes are predicted then execution continues to decision steps 4, 5, and 6 (Ratio between the two work episodes, break time duration, and work start time) then the process terminates.

7.3.2 Process Model 2

As shown in Figure 7.2, process model 2 is another representation of the decision steps that constitute the work activity process model. As an essential model (decision step), the work activity model is the first model in the process model. The process then executes the work activity duration if a work activity is predicted. Unlike process model 1, in process model 2 the work duration start time is then executed. Where the number of work episodes is then consulted. This shifts down the activation dependency at decision step 3 in process model 1

down. In addition, attributes interdependencies are modified, as the work start time model now contains less attributes (as indicated by the dashed arrow line). Further, if two work episodes are predicted, the decision steps 5 and 6 are executed, otherwise the process terminates.

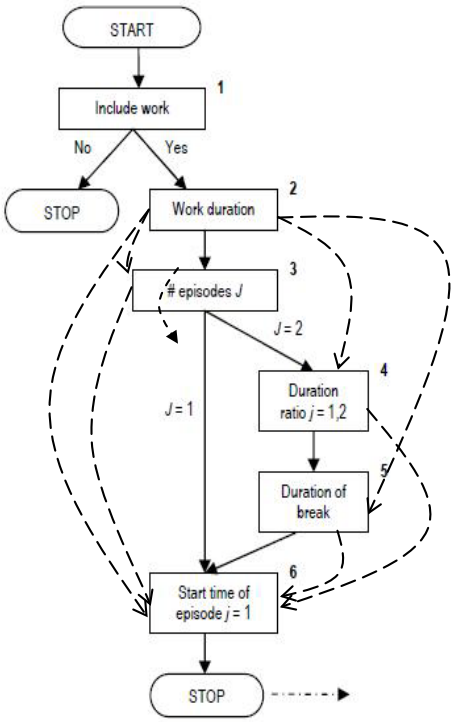


Figure 7.1 Process models 1: The original work activity process model attribute inclusion diagram in ALBATROSS

This sequence is developed and adapted because shifting the more work episodes down and executing the work activity start time before it, implies that the work start time attribute will be added to the attribute list of the model. And this may enhance the predictive performance of the more work episodes. As concluded in the Chapter 6, given that the added attribute is relevant in predicting the class attribute.

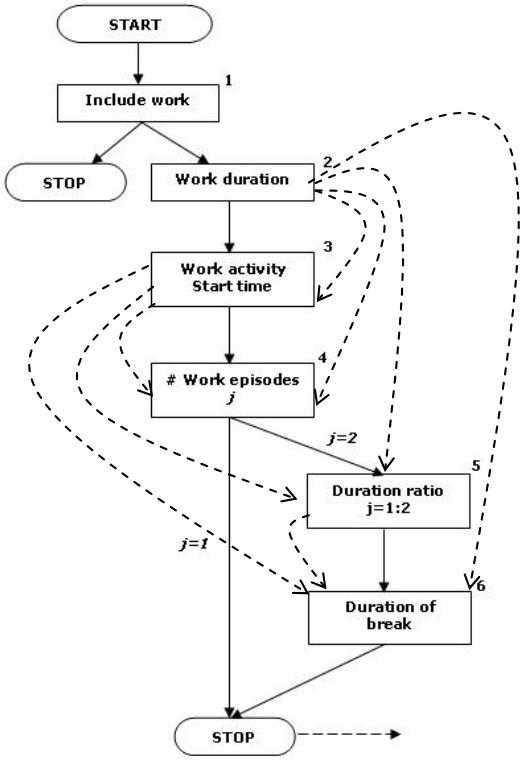


Figure 7.2 Process models 2: work activity process model attribute inclusion diagram in FEATALB

7.3.3 Process Model 3

In process model 3 (Figure 7.3), the number of work episodes is shifted upwards as decision step 2. After executing decision step 1 (work activity model), the number of work episodes are specified. If two work episodes are specified, then the ratio and break time models are consulted. Otherwise, if one work episode is predicted, the duration and start time models are executed consecutively. As indicated by the dashed arrow lines in Figure 7.3 the decision outcomes that are included in the duration model (decision step 5) is more in process model 3. The data set for the model in decision step 2 (more work episodes) contains less attributes than in the other decision processes.

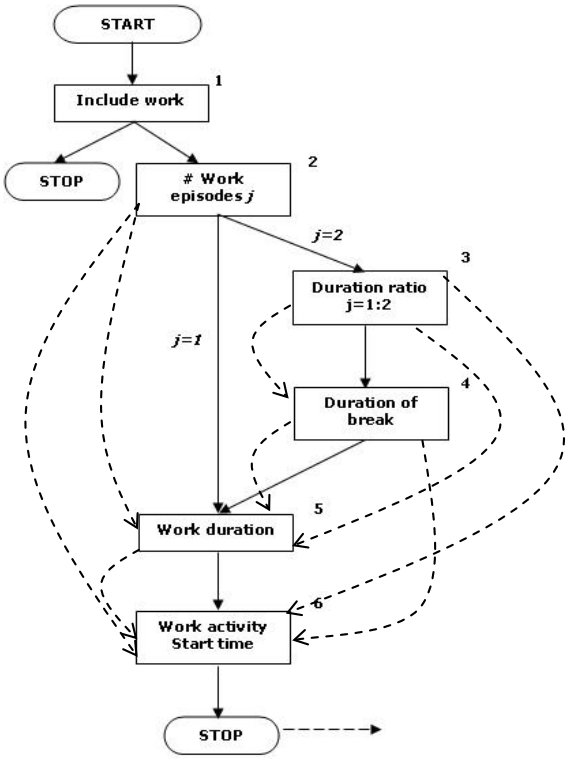


Figure 7.3 Process models 3: work activity process model attribute inclusion diagram in FEATALB

7.3 Analysis and Results

The ALBATROSS system applies a set of rules (if-then rules), which forms the scheduling process model. (Arentze and Timmermans 2004). These rules are accommodated in the rule-based scheduling engine to infer individuals' activity schedules at the household level. In ALBATROSS, the work activity process model contains six decision steps that are used to predict work activity related information such as, work activity duration, start time, break time if two work activity episodes are predicted. At each decision step, the decision models are extracted from activity diary data. Moreover, the process model contains the

attribute dependency feature, which denotes the inclusion of decision outcomes as attributes in subsequent decision steps. Hence, for each proposed process model, i.e. process model 2 and process model 3, the data sets are modified according to their order in the sequence. Modification of the data sets involved adding/deleting attributes to comply with the sequence of decisions in each process model. The results are presented separately for each process model, the original work activity process model which will be referred to as process model 1, process model 2 and process model 3.

The discrete choice decision steps are critical decision steps, because at these decision steps the activation dependency feature is enforced. It is noteworthy that for all work activity process models presented in this chapter decision step 1 is the same (include work activity), since it is the first in the sequence. Therefore, at the individual classifier accuracy validation level decision step 1 is the same. The second discrete choice decision step is the number of work episodes (decision step 3 in process model 1, decision step 4 in process model 2, and decision step 2 in process model 3) affects the execution of decisions in the work activity process models.

To serve the purpose of the analysis, two induction methods are used for discrete choice models, namely C4.5 decision trees and Logit models. While for continuous choice decision steps CART models are trained and employed. This implies that for each process model the results for C4.5 with CART decision trees and Logit with CART are presented separately.

In the next subsections, the validations of models of each process model are presented at the three validation levels. Firstly, the individual classifier level, then the activity pattern level and thirdly, at the work activity OD matrix and work start time (spatial and temporal) level.

7.3.1 Classifier Level Accuracy Analysis

The evaluation criteria of the discrete choice models are presented using two accuracy measures, the confusion matrix (also called contingency table) accuracy measure, since both discrete choice classifiers are binary. And the

Brier score (Brier 1950) because of the probabilistic action assignment rule used in scoring the models. The continuous choice models are validated using the Relative Absolute Error (RAE). RAE provides a measure of how good a predicted value is relative to the observed value. The reason for selecting this measure is that it can be reported as a percent error measure for numeric or continuous predictions. The RAE is calculated by dividing the sum of the absolute difference between the predicted and observed values by the observed cases.

Discrete choice models

For all process models, decision step 1 (include work activity) was set to be the first step, since the remaining decision steps models are related to the work activity. Therefore, the accuracy statistics for decision step 1 and for the C4.5 and Logit models is the same for all work activity process models introduced in this chapter. Table 7.1 summarizes the results of the analysis to assess model performance of decision step 1 for the three process models.

The results suggest that the Logit model outperforms the C4.5 specially in predicting the positive class value (1). The predictive performance (sensitivity) for the (*1*) class variable, which is the minority class, is notably higher in the Logit method. This can be explained by the fact that Logit is known to often outperform decision tree approaches for small size datasets (King and Zeng 2001). The F-measure, which is the measure of the dropout class taking into consideration sensitivity and precision, the Logit model is significantly higher than the C4.5 model.

Decision step - 1 Work	Training set			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.113781	0.813008	0.839448	0.73026
C4.5	0.114957	0.59248	0.84497	0.594898
Decision step - 1 Work	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.115959	0.791045	0.83628	0.704653
C4.5	0.115108	0.556503	0.825519	0.545455

Table 7.1 Accuracy statistics for decision step 1 (classifier level) for process models 1, 2 & 3

In Table 7.2 the accuracy statistics for the baseline, target, C4.5, and Logit models are reported. The results show that the M-IFN and the fully-informed approaches represent the baseline and target predictive performance measures respectively. The sensitivity and F-Measure of the target approach is the highest among all approaches. The Logit model reported performance because this is the first model (decision step 1) in the process model and no attributes are added to this model. While the M-IFN approach reported the lowest accuracy measures.

Approach	1-Brier Score	Sensitivity	Specificity	F-Measure
Baseline	0.72	0.30	0.85	0.22
Logit	0.88	0.81	0.84	0.73
C4.5	0.88	0.56	0.83	0.54
Target	0.88	0.81	0.84	0.73

Table 0.2 Baseline, C4.5, Logit and target accuracy statistics for decision step 1 (classifier level) for process models 1, 2 & 3

Table 7.3 shows the accuracy statistics for decision step 3 (number of work episodes). For this model the C4.5 model also outperform the Logit model. The reason for the weaker performance of the Logit approach is that the data set at decision step 3 is highly skewed (87%). This leads to an underestimation of the rare class calculated by Equation 3.8 as reported by King and Zeng (2001).

Decision step - 3 More_Work_Ep	Training set			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.116217	0.078125	0.967328	0.120482
C4.5	0.11485	0.171875	0.872812	0.169884
Decision step - 3 More_Work_Ep	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.156222	0.038961	0.961832	0.0631579
C4.5	0.145401	0.181818	0.862595	0.193103

Table 7.3 Accuracy statistics for decision step 3 for process models 1

The results reported in tables 7.1 and 7.3 conform to those proposed in the literature. For small size datasets Logit outperforms decision tree induction methods. Furthermore, Logit models perform poor for highly unbalanced datasets (Lim et. al., 2000).

Table 7.4 shows a comparison between baseline and target approaches with C4.5 model for process model 1. The Logit model is rejected because it fails to compete with C4.5 model for decision step 3. The results reveal that the predictive performance of the C4.5 method, in process model 1, is better than the baseline approach. And it did not reach the upper performance bound of the target approach discussed in Chapter 6.

Model	1-Brier Score	Sensitivity	Specificity	F-Measure
Base line	0.86	0.12	0.85	0.11
C4.5	0.85	0.18	0.86	0.19
Target	0.87	0.19	0.90	0.23

Table 7.4 Baseline, C4.5, and target accuracy statistics for decision step 3 for process models 1

For process model 2, the accuracy statistics and predictive performance of decision step 4 (number of work episodes) is shown in Table 7.5. The results show that as expected the predictive performance of the C4.5 model is better than the Logit model, since the dataset for this model is highly unbalanced.
The C4.5 model in process model 2 performs better than C4.5 model in process model 1, as reported by the F-Measure. In addition, considering the Sensitivity of the C4.5 model in process model 2 reveals an increase in the accuracy of predicting the positive class (the minority class).

Decision step - 4 More_Work_Ep	Training set			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.234876	0.189063	0.784131	0.211429
C4.5	0.101803	0.203125	0.908985	0.224138
Decision step - 4 More_Work_Ep	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.268773	0.289063	0.755725	0.188482
C4.5	0.147368	0.202857	0.882952	0.224179

Table 7.5 Accuracy statistics for decision step 4 (classifier level) - process models 2

This increase in accuracy can be explained by the attribute interdependency feature in the process model. As shown in figure 7.2 (the dashed lines arriving at decision step 4), the *number of work episodes* model in process model 2

contains an additional attribute (*work start time*) in the data set. Further to justify that this additional attribute increase the learning ability of the model, the data set is analysed using the Relief feature selection technique. The results from the Relief method is shown in table 7.6, which demonstrates that the new attribute (work start time) ranks the third relevant attribute in identifying the class variable.

Relief weight	Ranked attributes
0.12439	Dur
0.09553	Xdag
0.09502	Start_time
0.08923	Xarb
0.08811	Xn_dag

Table 7.6 Relief feature selection results for the number of work episodes in process model 2.

These results conform also with the results obtained in chapter 6, which concluded that attributes interdependencies between decision steps improves the performance of models in activity-based models.

Positioning the performance of the C4.5 method of process model 2 on the ranked performance scale (Table 7.7), the results show that the performance is approaching the target approach performance. Moreover, considering the sensitivity of the *C4.5 – Process model 2* indicate that it slightly performs better than the target approach in predicting the positive class. But overall (the F-Measure and Brier Score) the target approach performs best.

Model	1-Brier Score	Sensitivity	Specificity	F-Measure
Base line	0.86	0.12	0.85	0.11
C4.5 – Process model 1	0.85	0.18	0.86	0.19
C4.5 – Process model 2	0.85	0.20	0.88	0.22
Target	0.87	0.19	0.90	0.23

Table 0.7 Baseline, C4.5, and target accuracy statistics for decision step 3 for process models 1 and 2

Tables 7.8 show the accuracy statistics for the number of work episodes model (decision step 2) in process model 3. The *More_Work_Ep* decision model in process model 3 is the second decision step, where no additional attributes are added when the execution is reached. This indicates that the *More_Work_Ep*

model is the same as the *More_Work_Ep* model in the non-informed approach presented in Chapter 6. Therefore, it is expected to have similar performance. The accuracy statistics (Table 7.8) show that the C4.5 performs better than the Logit model. Moreover, comparing this with the accuracy results of process models 1 and 2 show that both models outperform the model in process model 3 (Table 7.9). Although process model 3 is process model dependent approach but the attribute interdependency feature has proven to enhance the predictive performance of individual models.

Decision step - 2 More_Work_Ep	**Training set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.113867	0.0625	0.994166	0.111475
C4.5	0.11308	0.109375	0.873979	0.112
Decision step - 2 More_Work_Ep	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Logit	0.146908	0.012987	0.982188	0.0235294
C4.5	0.138436	0.142857	0.882952	0.164179

Table 7.8 Accuracy statistics for decision step 2 (classifier level) for process models 3

Model	**1-Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Base line	0.86	0.12	0.85	0.11
C4.5 – Process model 3	0.86	0.14	0.88	0.16
C4.5 – Process model 1	0.85	0.18	0.86	0.19
C4.5 – Process model 2	0.85	0.20	0.88	0.22
Target	0.87	0.19	0.90	0.23

Table 0.9 Baseline, C4.5, and target accuracy statistics for decision step 3 for process models 1, 2, and 3

The reason for this behavior is that for process model 1 the duration model is added as attribute to the *More_Work_Ep* model. And for process model 2 the duration and start time attributes are added. In addition, running the Relief feature selection method on this decision model shows that both attributes are relevant in predicting the class attribute. The results also show that for decision tree models when more attributes are added to the model the probability of predicting the positive class (minority class) increases, i.e. the Sensitivity measure. Table 7.10 summarizes the Sensitivity measure for the test set for all three process models.

More_Work_Ep	Process Model 1	Process Model 2	Process Model 3
Logit	0.04	0.29	0.01
C4.5	0.18	0.20	0.14

Table 7.10 Sensitivity accuracy measure for all process models

The sensitivity measure approximates the probability of the positive class being correctly classified. In the case of the *More_Work_Ep* model the positive class (two work episode) is the rare class. As discussed above, the results (Table 7.10) show that in general C4.5 performed better than Logit models in predicting the positive class (yWo) except for process model 2. The Logit model in process model 2 obtained better *More_Work_Ep* model than all other models in all process models with sensitivity at 0.289 against 0.039 and 0.013 for process models 1 and 2 respectively. Although the dataset for this model is highly unbalanced, adding an extra attribute (*Start time*) enhances the predictive performance of the rare class for the Logit model. This means that, in the Logit model, the *start time* attribute has a high effect on the predictor (*More_Work_Ep*). In conclusion for discrete choice models, it is recommended to use Logit model for decision model 1 in all process models. In addition, considering process model 1, it is recommended to use the C4.5 model for decision model 3. Process model 2, in predicting the rare class (2 work episodes) performs best when using the Logit method for all discrete choice models. However, when considering the overall performance, it is also recommended to use the C4.5 model for decision model 3. For decision model 3, the Logit model does not perform well at decision model 3 and thus, the C4.5 model is preferable.

Continuous choice models

CART decision trees are used in all continuous decision models for all process models. The models are evaluated using the RAE measure. The results of the models are shown in Tables 7.11, 7.12 and 7.13 for process models 1, 2 and 3 respectively. Considering the results, we notices that all models in all decision steps and for all process models reported approximately similar RAE. This can be explained by the numeric nature of the models and the derivation of rules of

continuous models adapted in ALBATROSS, which assumes a normal distribution at each leaf node in continuous choice decision trees.

Decision Step	Name	Training set	Test set
2	**Work_Dur**	19%	20%
4	**Ratio**	20%	19%
5	**Break_Time**	52%	65%
6	**Begin_Time**	8%	10%

Table 7.11 RAE for CART continuous choice classifiers for process model 1

Decision Step	Name	Training set	Test set
2	**Work_Dur**	19%	20%
5	**Ratio**	18%	22%
6	**Break_Time**	52%	66%
3	**Begin_Time**	10%	11%

Table 7.12 RAE for CART continuous choice classifiers for process model 2

Decision Step	Name	Training set	Test set
5	**Work_Dur**	20%	20%
3	**Ratio**	19%	23%
4	**Break_Time**	52%	71%
6	**Begin_Time**	8%	10%

Table 7.13 RAE for CART continuous choice classifiers for process model 3

The RAE for all continuous models for process models 1, 2, and 3 and for the baseline and target approaches are shown in Table 7.14. It is noted that an increase in the RAE is reported for the work duration model (decision step 5) in process model 3. Although additional attributers are added to this model (attribute interdependency feature), which proved to improve the predictive performance of individual decision step models. But the work duration model in process model 3 reported about the same RAE. To further understand this plight, the dataset for this model is analysed using the Relief feature selection technique. The results reveal that none of the added attributes ranked among the first 10 most relevant attributes. In addition, the CART decision tree for this model is analysed by investigating the attributes that are selected for splitting of

nodes. And it was revealed (Figure 7.4) that additional attributes, which result from changing the order of this decision step, are not active (not used for splitting of the decision tree).

Model	Baseline	Process Model 1	Process Model 2	Process Model 3	Target
Work_Dur	26%	20%	20%	20%	20%
Ratio	31%	19%	22%	23%	20%
Break_Time	79%	65%	66%	71%	61%
Begin_Time	12%	10%	11%	10%	10%

Table 0.14 Continuous models test sets RAE

```
<TreeModel modelName="RPart_Model" functionName="regression"
 <MiningSchema>
  <MiningField name="Work_dur" usageType="predicted"/>
  <MiningField name="Urb" usageType="supplementary"/>
  <MiningField name="Comp" usageType="active"/>
  <MiningField name="Child" usageType="supplementary"/>
  <MiningField name="Day" usageType="active"/>
  <MiningField name="pAge" usageType="supplementary"/>
  <MiningField name="SEC" usageType="supplementary"/>
  <MiningField name="Ncar" usageType="supplementary"/>
  <MiningField name="Gend" usageType="active"/>
  <MiningField name="Driver" usageType="supplementary"/>
  <MiningField name="wstat" usageType="supplementary"/>
  <MiningField name="Pwstat" usageType="supplementary"/>
  <MiningField name="Xdag" usageType="active"/>
  <MiningField name="Xn_dag" usageType="active"/>
  <MiningField name="Xarb" usageType="supplementary"/>
  <MiningField name="Xpop" usageType="supplementary"/>
  <MiningField name="Ddag" usageType="supplementary"/>
  <MiningField name="Dn_dag" usageType="supplementary"/>
  <MiningField name="Darb" usageType="supplementary"/>
  <MiningField name="Dpop" usageType="supplementary"/>
  <MiningField name="Empty" usageType="supplementary"/>
  <MiningField name="Empty1" usageType="supplementary"/>
  <MiningField name="Nep" usageType="supplementary"/>
  <MiningField name="Ratio" usageType="supplementary"/>
  <MiningField name="Inter" usageType="supplementary"/>
 </MiningSchema>
```

Figure 7.4 Mining schema for the work duration model (decision step 5) PMML CART decision tree in process model 3.

Figure 7.4 illustrates a snapshot from the decision tree PMML model's MiningSchema section, which describes the attributes names and usage types (usageType). According to the PMML XML schema, the usageType field can contain three values (Guazelli et al. 2009):

- **"active"**: field used as input (independent field).
- **"predicted"**: field whose value is predicted by the model.
- **"supplementary"**: field holding additional descriptive information. Supplementary fields are not required to apply a model. They are provided as additional information for explanatory purpose, though. When some field has gone through preprocessing transformations before a model is built, then an additional supplementary field is typically used to describe the statistics for the original field values.

The *Start time* model reported approximately similar RAE in all process models, with a slight increase in favour of process model 2. Since it has moved up in the process model, which implies that its data set contained less attributes (decision outcomes).

Considering the *Break time* model, the RAE reported lowest in process model 1, and around the same RAE in process models 2. The slight increase in the RAE in process model 3 is because its data set contains less attributes.

Finally the *Ratio* model obtained less RAE in process model 1 and highest in process model 3. The reason for the RAE increase in process model 3 is because no decision outcomes are added. Furthermore, in process model 2, where the RAE is expected to decrease since an extra attribute (*Start time*) is added to the data set, but it is noted that the RAE has increased. Assessing the PMML model for this model, the extra is marked as supplementary and so it is not used in the splitting of the decision tree model.

7.3.2 Activity Pattern Level

The second validation level for the three process models is assessed by measuring how similar are observed with predicted activity pattern sequences. The SAM measure is used for this purpose since it works by calculating the effort

(distance) needed to equalize two sequences. It is important to note that the analysis in this thesis is performed on the work process model only. Hence, the SAM distance is calculated for all activities sequences and for work activity sequences for each process model's schedule outcome. The lower the SAM distance the closer the predicted activity patterns to the observed ones. The confusion matrices for observed versus predicted sequences lengths for process models 1, 2, and 3 are presented in Tables 7.15, 7.16, and 7.17 respectively. The confusion matrices for process model 1 show that the C4.5 model is able to predict work activity sequences lengths that are closer to the observed ones. Except for sequences with two symbols, the Logit model is more accurate. For process models 2 and 3, the confusion matrices show that for work activities with one symbol (no work activity) are comparable. However, the Logit model is more accurate in predicting work sequences with two and four symbols.

	Training set	Predicted					Test set	Predicted			
C4.5	Sequence Length	1	2	4	Total	C4.5	Sequence Length	1	2	4	Total
Observed	1	0.80	0.17	0.04	0.52	Observed	1	0.82	0.17	0.01	0.52
	2	0.39	0.54	0.07	0.36		2	0.41	0.49	0.10	0.32
	4	0.46	0.46	0.09	0.12		4	0.35	0.52	0.13	0.16
	Total	0.61	0.33	0.05	589.00		Total	0.61	0.33	0.06	298.00
	Training set	Predicted					Test set	Predicted			
Logit	Sequence Length	1	2	4	Total	Logit	Sequence Length	1	2	4	Total
Observed	1	0.73	0.25	0.02	0.52	Observed	1	0.72	0.28	0.00	0.52
	2	0.12	0.86	0.02	0.36		2	0.09	0.90	0.01	0.32
	4	0.10	0.84	0.06	0.12		4	0.08	0.88	0.04	0.16
	Total	0.44	0.54	0.02	589.00		Total	0.42	0.57	0.01	298.00

Table 7.15 Process model 1 confusion matrix for work activity sequences length

	Training set	Predicted					Test set	Predicted			
C4.5	Sequence Length	1	2	4	Total	C4.5	Sequence Length	1	2	4	Total
Observed	1	0.78	0.19	0.03	0.52	Observed	1	0.79	0.18	0.03	0.52
	2	0.43	0.52	0.05	0.36		2	0.36	0.56	0.07	0.32
	4	0.47	0.44	0.09	0.12		4	0.29	0.65	0.06	0.16
	Total	0.62	0.34	0.04	589.00		Total	0.57	0.38	0.05	298.00
	Training set	Predicted					Test set	Predicted			
Logit	Sequence Length	1	2	4	Total	Logit	Sequence Length	1	2	4	Total
Observed	1	0.73	0.18	0.09	0.52	Observed	1	0.72	0.23	0.05	0.52
	2	0.12	0.67	0.21	0.36		2	0.09	0.69	0.22	0.32
	4	0.10	0.68	0.22	0.12		4	0.08	0.79	0.13	0.16
	Total	0.44	0.41	0.15	589.00		Total	0.42	0.47	0.11	298.00

Table 7.16 Process model 2 confusion matrix for work activity sequences length

	Training set	Predicted					Test set	Predicted			
C4.5	Sequence Length	1	2	4	Total	C4.5	Sequence Length	1	2	4	Total
Observed	1	0.79	0.19	0.02	0.52	Observed	1	0.79	0.21	0.01	0.52
	2	0.40	0.54	0.06	0.36		2	0.34	0.60	0.05	0.32
	4	0.41	0.56	0.03	0.12		4	0.35	0.56	0.08	0.16
	Total	0.60	0.36	0.03	589.00		Total	0.57	0.39	0.03	298.00
	Training set	Predicted					Test set	Predicted			
Logit	Sequence Length	1	2	4	Total	Logit	Sequence Length	1	2	4	Total
Observed	1	0.73	0.17	0.09	0.52	Observed	1	0.72	0.23	0.05	0.52
	2	0.12	0.66	0.23	0.36		2	0.09	0.67	0.24	0.32
	4	0.10	0.68	0.22	0.12		4	0.08	0.75	0.17	0.16
	Total	0.44	0.40	0.16	589.00		Total	0.42	0.46	0.13	298.00

Table 7.17 Process model 3 confusion matrix for work activity sequences length

The SAM distances between observed and predicted activity pattern sequences are shown in Tables 7.18, 7.19 and 7.20. As discussed in chapter 6, for process model 1 (the original process model) the C4.5 model obtained lower SAM distances for all activities and work activity patterns. The results also reveal that

the SAM measure for process model 1 for all activities and work activity patters are lower than the SAM measure for the other two process models.

Considering the SAM distances for all activity patterns, the SAM distances are approximately the same for process models 2 and 3. And in general the C4.5 models reported lower SAM distances than the Logit models.

	SAM distance all activities		SAM distance work activities	
	Logit	**C4.5**	**Logit**	**C4.5**
Training	4.51	4.12	0.59	0.53
Test	4.35	3.94	0.59	0.52

Table 0.18 All activities and work activity sequences SAM distances - process model 1

	SAM distance all activities		SAM distance work activities	
	Logit	**C4.5**	**Logit**	**C4.5**
Training	4.68	4.64	0.58	0.59
Test	4.61	4.65	0.58	0.59

Table 7.19 All activities and work activity sequences SAM distances - process model 2

	SAM distance all activities		SAM distance work activities	
	Logit	**C4.5**	**Logit**	**C4.5**
Training	4.64	4.54	0.82	0.55
Test	4.81	4.63	0.81	0.57

Table 7.20 All activities and work activity sequences SAM distances - process model 3

For the work activity pattern, the SAM distances of the C4.5 models are lower than those of the Logit models for all process models. Process model 1 obtained better SAM distance at 0.53, than process models 2 and 3 at 0.59 and 0.55 respectively for training sets. And 0.52 against 0.59 and 0.57 for test sets. The SAM distance for the Logit models of process models 1 and 2 are almost similar (0.59 and 0.58) and lower than the SAM distance reported in process model 3 at 0.81 for the training set. This can be explained because the number of work episodes (decision step 2) in process model 3 is trained without the extra features (attribute dependency feature) and so, mainly predicts one work activity. This means, on average more effort is needed to equalize the predicted and observed work activity sequences.

	Observed	Baseline	Process model 1	Process model 2	Process model 3	Target
All activities	5.02	3.3	3.8 (C4.5)	3.7 (C4.5)	3.5 (C4.5)	3.94 (C4.5)
Work Activities	1.7	1.2	1.5 (C4.5)	1.4 (C4.5)	1.55 (C4.5)	1.5 (C4.5)

Table 7.21 Mean length of all and work activities

Table 7.12 shows the lengths of sequences for the baseline and target approaches compared to sequences lengths of process models 1, 2, and 3. The results reveal that sequences lengths are comparable, having the M-IFN approach as the baseline model. Finally, the validation at the activity patterns shows that changing the order of decision steps, while maintaining attributes interdependencies, does not affect the predictive performance. In conclusion, as measured by SAM, the validation at the activity pattern level, the SAM distance cannot detect local changes (changing the order of decision models) in the process model.

7.3.3 Work Activity Trip Matrix and Trips Start Time Level Accuracy Analysis (Spatial and Temporal Resolutions)

At this validation level, the observed and predicted work trip origin-destination (OD) matrices are compared via a correlation coefficient. In addition, a correlation coefficient is evaluated between observed and predicted work trips start times. The correlation coefficient measures the correspondence between the observed and predicted number of trips. Tables 7.22, 7.23 and 7.24 illustrate the performance of the training and test sets of process models 1, 2 and 3 respectively.

The results at the work activity trip matrix level, the three process models reported comparable correlation coefficients. As reported in previous chapters, C4.5 models outperformed Logit models, and this applies for all process models. C4.5 models obtained 0.84, 0.83, and 0.82 correlation coefficients in the training sets against 0.81, 0.81, and 0.82 in Logit models for process models 1, 2, and 3 respectively.

	Work activity trip matrix level		Work activity start time per hour of the day	
Dataset	**Logit**	**C4.5**	**Logit**	**C4.5**
Training	0.81	0.84	0.87	0.90
Test	0.80	0.82	0.77	0.80

Table 7.22 Work activity trip matrix and start time per hour of the day correlation coefficients for process model 1

	Work activity trip matrix level		Work activity start time per hour of the day	
Dataset	**Logit**	**C4.5**	**Logit**	**C4.5**
Training	0.81	0.83	0.87	0.90
Test	0.80	0.84	0.83	0.92

Table 0.23 Work activity trip matrix and start time per hour of the day correlation coefficients for process model 2

	Work activity trip matrix level		Work activity start time per hour of the day	
Dataset	**Logit**	**C4.5**	**Logit**	**C4.5**
Training	0.82	0.82	0.85	0.90
Test	0.80	0.83	0.81	0.91

Table 0.24 Work activity trip matrix and start time per hour of the day correlation coefficients for process model 3

The work activity start times were extracted from predicted output schedules for each process model. Then the correlation coefficients between observed and predicted work activity start times were calculated. The results shown in Tables 7.12, 7.13 and 7.14 illustrates that the three process models reported about the same performance for the two techniques (C4.5 and Logit). The C4.5 outperformed the Logit model in all process models with 0.9, opposed to around 0.87 for Logit models in the training sets. While for the test sets C4.5 reported 0.8, 0.92, and 0.91 and 0.77, 0.83, and 0.81 in Logit models for process models 1, 2 and 3 respectively.

Table 7.25 summarises the correlation coefficient for baseline, target, process model 1, process model 2, and process model 3 for work activity trips and work activity start time. All approaches reported approximately similar results for the work activity trips. The validation of the models at this level, reveal that at this aggregated level is not able to detect local changes i.e. changing the order of decision steps or training the models independently.

	Baseline	Process model 2	Process model 2	Process model 3	Target
OD Work activity trips	0.83	0.82	0.84	0.83	0.82
Work activity start time	0.82	0.8	0.92	0.91	0.85

Table 7.25 Spatial and temporal correlation coefficients

7.4 Conclusions

Computational process rule-based activity-based models are based on a set of rules which constitute the scheduling process of the simulation model. In ALBATROSS as a computational rule-based approach to modeling activity-travel patterns use 26 decision steps. This sequence of decision rules forms the scheduling engine which is the core of the ALBATROSS system. Each decision rule, might utilize an induction method, such as, a decision tree, Logit models, or any other induction method that is capable of predicting values as rules. As a result, the scheduling process model together with the decision rules models generate a feeling of a black box. In which the system is viewed solely in terms of its input, output, and individual decision rules (steps).

To gain full understanding of the ALBATROSS scheduling engine and its processes, the work activity process model is investigated. In addition to the original work activity process model in ALBATROSS, two work activity process models are developed in the context of the FEATALB framework. The newly developed process models (data representation) are implemented through the disposition of decision steps or rules within the process model. Moreover, two induction methods are utilized for discrete choice decision rules, namely C4.5 and Logit models. While the Classification and Regression Trees (CART) was employed for continuous choice decision rules.

The analysis was conducted by running simulations using each process model and for each induction method. Furthermore, validation of models for all process models is performed at the individual decision rule level, at the activity pattern sequence level, and at the work trip OD and work activity start times level. Developing other process models resulted in shifting the occurrence of the

activation dependency feature within process models. Additionally, the datasets used for training the models in decision steps' models were modified due to the attribute dependency feature.

The results of the analysis conclude that using the original process model (process model 1) it is recommended to use the Logit model for the first decision step (*Work*). At the individual classifier level validation it was revealed that the Logit model obtained the highest sensitivity and F-Measure. While for the third decision model (*More_Work_Ep*) the Logit model proved to be an outcast. In addition, using the C4.5 decision tree method, results in improving the predictive performance of the model. Consequently, for decision step 1 (*Work*) in process model 2, validation results at the individual classifier level, show that it is also preferable to use the Logit method. However, the Logit method is also an outcast for the More_Work_Ep model. Therefore, using the C4.5 method, results in better predictive performance. Process model 3 reported slightly weaker performance than other process models.

The validation results of the models were consistent to the results reported in previous chapters with regard to the attribute interdependencies between decision steps. However, another factor that affects the predictive performance of process models is the relevance of the decision outcome that is added in subsequent decision models in predicting the class variable. In general process models with C4.5 models outperformed process models simulated with Logit models. The results also suggest that the disposition of decision steps or experiment other data representations within the work activity process model does not lead to significantly improve the predictive performance of the model. This is confirmed by the validation at the aggregated levels (activity pattern and Spatial and temporal levels). Hence, using the currently implemented work activity process model can achieve satisfactory work activity schedules.

References

Arentze, T.A., and Timmermans, H.J.P. A Learning-Based Transportation Oriented Simulation System. Transportation Research Part B, 2004, 38, 613-633.

Arentze, T. A., and H. J. P. Timmermans (2005). Albatross 2: A Learning-Based Transportation Oriented Simulation System. European Institute of Retailing and Services Studies, Eindhoven, Netherlands.

Brier, G. W. Verification of forecasts expressed in terms of probability. Mon. Wea. Rev., 1950, 78, 1–3.

Ettema, D. F., A. W. J. Borgers, and H. J. P. Timmermans. Using Interactive Computer Experiments for Identifying Activity Scheduling Heuristics. Presented at 7th International Conference on Travel Behavior,
Santiago, Chile, 1994.

Ettema, D., and H. Timmermans. (2003) Modeling Departure Time Choice in
the Context of Activity Scheduling Behavior. In Transportation Research Record: Journal of the Transportation Research Board, No. 1831, Transportation Research Board of the National Academies, Washington, D.C.,
2003, pp. 39–46.

Guazelli, Alex, Wen-Ching Lin and Tridivesh Jena. (2010) PMML in Action: Unleashing the Power of Open Standards for Data Mining and Predictive Analytics. CreateSpace. 2010, ISBN 978-1452858265, pp 5.

Harry Timmermans, Theo Arentze and Chang-Hyeon Joh (2002), Analysing space-time behaviour: new approaches to old problems, Progress in Human Geography 26,2 (2002) pp. 175–190

King, G. and Zeng, L. (2001). Logistic regression in rare events data. Political Analysis, 9:137-163.

Lim, T.S., Loh, W.Y. and Shih, Y.S. (2000) A Comparison of Prediction Accuracy, Complexity, and Training Time for Thirty-three Old and New Classification Algorithms" Machine Learning, 40, 203-228.

Ruiz, T., and Roorda, M. J., (2008). Analysis of Planning Decisions During the Activity-Scheduling Process. Transportation Research Record: Journal of the Transportation Research Board, No. 2054, Transportation Research Board of the National Academies, Washington, D.C., pp. 46–55.

Williams, G. J. Rattle: A Data Mining GUI for R. The R Journal, 1(2), 2009, 45-55.

Chapter 8

Sensitivity Analysis of process models in FEATALB framework

In previous chapters, the work activity process model was experimented and analysed in an attempt to obtain better performance. This was performed using two approaches, acquiring better classification models and by obtaining better data representation. In both approaches, the decision tree models trained at each decision steps were generated by setting up the minimum cases at leaf nodes at 30 cases. This minimum number of cases was chosen to avoid over-fitting. In addition, the stochastic approach for the action assignment rule used in ALBATROSS was also adapted throughout the experiments and analyses conducted. The results obtained revealed that for some steps decision tree models outperformed Logit models. In this chapter, a sensitivity analysis of the decision tree models will be performed to assess the performance of the FEATHERS / ALBATROSS model. First, the simulation model is run using a deterministic action assignment rules at leaf nodes rather than the stochastic approach. Second, the decision tree models will be trained while setting the minimum number of cases at leaf nodes to lower than 30 cases. This allows for evaluating the predictive performance of the model and test whether the model is sensitive to changing such parameters or not for the Flemish data.

This chapter is structured as follows. In the next section, the main concepts of deterministic and probabilistic action assignment rules are discussed. In addition, issues related to decision tree learning, such as setting the minimum number of cases at leaf nodes and its effect on model performance, over-fitting and under-fitting are explained. Followed by experimenting the deterministic action assignment rules in the models and validation results are discussed. Then the analysis of how sensitive the models of the work activity process model are to the minimum number of cases at leaf nodes. Finally the chapter ends with conclusion and discussion.

8.1 Introduction

The statistical literature on decision tree classification focuses on data segmentation and recognition of interactions between variables prior to modeling. In addition, attention has been paid to methods of validation of resulting classification models. However, deriving rules from decision trees (also called action-assignment rules) that are used for deriving predictions from a decision tree model is usually deterministic. The deterministic action assignment rules use the plurality rule. The plurality rule records for each leaf node the modal action in the training set and classify this action to each new case at that node (Rasouli et. al, 2011). In their study, Arentze and Timmermans (2003) argued that deterministic action assignment rules do not meet the requirements of activity-based travel behavior. Deterministic rules do not reproduce residual variances after fitting the data, and as a result tend to generate biased choice distributions at the aggregate level, especially for skewed data sets. Therefore, they developed a probabilistic action assignment rules for discrete choice and continuous decision tree models (Arentze and Timmermans, 2003). However, the proposed probabilistic action assignment rules did not solve the problem of producing bias free predictions. The analyses in previous chapters were conducted using the probabilistic action assignment rules developed for ALBATROSS using the Flemish data i.e. using the FEATALB framework. To this end, the sensitivity of the model is appraised by applying a deterministic action assignment rule for decision tree (discrete choice and continuous) models. The analysis is performed using C4.5 decision tree models at the six decision steps in the process model. All models are trained with pruning and setting the minimum cases at leaf nodes to 30 cases, i.e. using the same C4.5 decision tree models used in Chapter 6.

Another issue of concern to perform a sensitivity analysis is the minimum number of case at leaf nodes (30 in ALBATROSS). However, there is no clear argument, why it was set to 30, except to avoid over-fitting as argued by Arentze and Timmermans (2005). The minimum number of cases per leaf node is an

important option in decision tree training. This option allows forcing the lowest number of instances that can constitute a leaf node. The higher this number the more general the tree. So lowering this number will produce a more specific tree that fits the cases in the training set. On the other hand, increasing the minimum number of cases per leaf node produces more generalized trees. Nevertheless, obtaining a preferred number is hard to achieve and usually depends on the distribution of the class variable.

Therefore, in this chapter the decision tree parameters i.e. the minimum number of instances at leaf nodes is experimented to attempt to fine tune the models to obtain a preferred value. It is noteworthy that by increasing the minimum number of cases up to a certain level results in model under-fitting. Under-fitting generates models that are too simple so both training and test errors are large.

8.2 Deterministic Action Assignment Rules

As discussed in Chapter 3 (section 3.4) the action assignment rules, for discrete and continuous models, developed in ALBATROSS suggest probability distribution between classes. This probabilistic action assignment rule was developed by Arentze and Timmermans (2003) to serve the purposes of generating bias free predictions and consider the effects of space-time constraints of activity-based models. Hence, the developed probabilistic action assignment rule takes the space-time decision constraints into account but do not solve generating bias free predictions (Arentze and Timmermans, 2003). The same approach (probabilistic action assignment rules) is used in the FEATALB framework to predict schedules for the Flemish population. For the purpose of the sensitivity analysis of the work activity process model a deterministic action assignment rules for discrete and continuous models is applied. The models are compared with the models generated using the probabilistic approach. This allows for assessing the predictive performance of models at individual classifiers level and at aggregated level. And how sensitive are rule-based activity-based models to deterministic action assignment rules.

As discussed in the Introduction of this chapter, the deterministic action assignment rules approach use the plurality rule. The plurality rule implies that the group with the highest representation determines the class assignment at leaf nodes. To illustrate this, consider the C4.5 decision tree model for the work decision step (Figure 8.1). If a new case is to be classified with the *wstat* variable values is set to *0*. The model evaluates this variable at the root node going down to reach a leaf node (encircled in red) where a decision is made.

The information provided by this leaf node (*0 (1535.0 / 36.0)*), is read as follows, *0* is the predicted class, *1535.0* is the number of cases belonging to class *0* (no work) and 36.0 is the number of cases belonging to the opposite class (*1: work*). Hence, instead of using the probabilistic action assignment rule which calculates *0.98* as the probability of predicting class 0 and *0.02* for predicting class *1*, the predicted value will always be *0*.

Similarly for continuous decision trees, to predict a numeric value at a leaf node, the average value of the class variables at this leaf node is predicted.

The analysis is performed using C4.5 decision trees for discrete choice models and CART decision trees for continuous models.

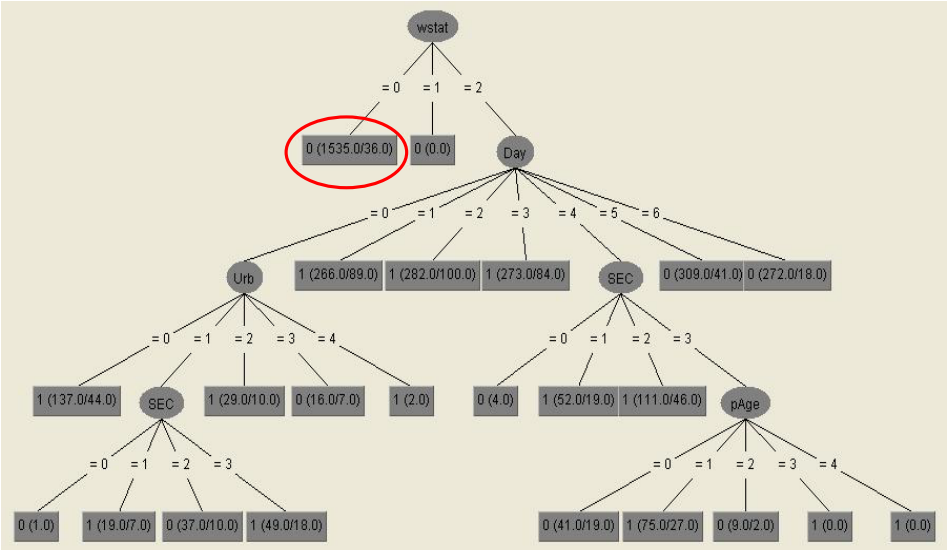


Figure 8.1 Work activity (decision step 1) C4.5 decision tree model.

The experiments were setup by running FEATHERS utilizing a decision tree model (C4.5 for discrete and CART for continuous models) at relevant decision steps. The model is validated at three levels, the individual classifier level, the activity pattern level, and the work activity trip matrix and work activity start time level. In the next subsections the model comparison criterion and validation results are discussed.

Classifier level validation

The validation results for discrete choice models are reported in Table 8.1. The results show that, for the *Work* model (decision step 1) using the deterministic approach, the predictive performance of predicting the positive class (1), which is the minority class has increased. This is detected by considering the Sensitivity measure. Sensitivity approximates the probability of the positive class being correctly classified. As reported in Chapter 7 (Table 7.1), the sensitivity of the training set for the work model, using the probabilistic approach, reported at 0.60 compared with 0.65 for the deterministic approach. As for the test set, the deterministic approach obtained 0.62 against 0.56 for the probabilistic action assignment rule. In addition, considering the F-Measure reveals that using the deterministic action assignment rules enhance the overall performance of the Work model. As shown in Table 8.1 the F-measure reported 0.75 and 0.73 for training and test sets respectively, compared to 0.60 for training set and 0.55 for the test set.

With regard to the more work episodes (More_Work_Ep) at decision step 3, it is noted that adapting the deterministic action assignment rules in the C4.5 model always predicts a one work episode (0). This can be explained because the data set for this model is highly skewed (87%) towards the one work episode class (0). Therefore, at leaf nodes the plurality rule dictates that the model always predict 0. This can spotted in the sensitivity statistic at 0 for both the training and test sets, while in the probabilistic action assignment rules the sensitivity of the model was at 0.24 and 0.19 for training and test sets respectively. The NA (Not Available) in the F-Measure indicates that the values cannot be computed since

the TP and FP values used to calculate the precision (Equation3.16) for this approach is zero.

Decision step - 1 Work	Brier Score	Sensitivity	Specificity	F-Measure
Training set	0.114957	0.646405	0.811834	0.747535
Test Set	0.115108	0.618976	0.805534	0.724508
Decision step - 3 More_Wrok_Ep	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
Training set	0.108202	0	1	NA
Test Set	0.134237	0	1	NA

Table 8.1 Deterministic discrete choice classifiers accuracy statistics for Work and More_Work_Ep models

Table (8.2) shows the accuracy of decision step 1 (*Work*) in the deterministic approach compared to the baseline and target approaches. We notice that the performance of the deterministic approach is better than the baseline approach. But the target approach still outperforms the deterministic approach. Fro decision step 3 the results (Table 8.3) show that adapting a deterministic action assignment rule results in an outcast. The results show that using a probabilistic action assignment rule for simulating activity-based models increase the predictive performance. This confirms the findings obtained by Arentze and Timmermans (2003).

Approach	1-Brier Score	Sensitivity	Specificity	F-Measure
Baseline	0.72	0.30	0.85	0.22
C4.5	0.88	0.62	0.81	0.72
Target	0.88	0.81	0.84	0.73

Table 0.2 Baseline, deterministic-C4.5, and target accuracy statistics for decision step 1

Model	1-Brier Score	Sensitivity	Specificity	F-Measure
Base line	0.86	0.12	0.85	0.11
C4.5	0.86	0	1	NA
Target	0.87	0.19	0.90	0.23

Table 0.3 Baseline, deterministic-C4.5, and target accuracy statistics for decision step 3

For continuous models, the validation results are shown in Table 8.4. The performance of continuous choice models was assessed by means of the RAE. As depicted in Table 8.4 results shows that for decision steps 2, 4 and 6, the RAE is 21, 22 and 10 % respectively, for training sets, and 22, 22 and 10 % for

test sets, while for decision step 5 the RAE reported 55% for training and 70% for test set. It is noted that these results are comparable with the RAE reported by the CHAID continuous decision trees for the stochastic approach as reported in Table 7.11. Furthermore, the deterministic approach outperformed the M-IFN approach for continuous models (Table 6.2), while in the target approach the RAE is still smaller. As shown in Table 8.5, the Baseline, deterministic and Target (stochastic)approaches are reported.

Decision Step	Name	Training set	Test set
2	**Work_Dur**	21%	22%
4	**Ratio**	22%	22%
5	**Break_Time**	55%	70%
6	**Begin_Time**	10%	10%

Table 8.4 RAE for deterministic continuous classifiers

Model	Baseline	Deterministic	Target
Work_Dur	26%	22%	20%
Ratio	31%	22%	20%
Break_Time	79%	70%	61%
Begin_Time	12%	10%	10%

Table 0.5 RAE for Baseline, Deterministic and target approaches.

Activity pattern level

The average length of all activities is 3.43 symbols and 1.42 symbols for work activities for the training set. While the average length of all activities is 3.45 and 1.44 for work activities for the test set. Using the deterministic action assignment rules, the C4.5 models predicted shorter activity patterns. Compared to 3.9 symbols for all work activities and 1.5 for work activities with the probabilistic rules. Recall that the average observed activity pattern length is 5.16 and the average observed work activity length is 1.7 symbols. In general, the deterministic approach predicted shorter activity patterns. This can be explained by the fact that at decision step 3 (More_Work_Ep) always predicted one work episode.

Table 8.6 shows the confusion matrices for the work activity sequences lengths of the deterministic approach. The results indicate that for the training set, the deterministic model predicted work sequences with one and two symbols only. This means that either no work activities are predicted or work activities with only one episode are predicted. And this explains the shorter work activity lengths.
The SAM distance (Table 8.7) for all activities is 4.5 and 4.68 for training and test sets respectively. As for the work activity pattern the SAM distance is 0.52 for the training set and 0.58 for the test set.

	Training set	Predicted			
C4.5	Sequence Length	1	2	4	Total
Observed	1	0.71	0.29	0.00	0.52
	2	0.09	0.91	0.00	0.36
	4	0.09	0.91	0.00	0.12
	Total	0.42	0.58	0.00	589.00

	Test set	Predicted			
C4.5	Sequence Length	1	2	4	Total
Observed	1	1.00	0.00	0.00	0.52
	2	1.00	0.00	0.00	0.32
	4	1.00	0.00	0.00	0.16
	Total	1.00	0.00	0.00	298.00

Table 8.6 Deterministic confusion matrix for work activity sequences length

	All Activities	Work Activities
Training	4.51	0.52
Test	4.7	0.58

Table 0.7 Deterministic SAM distance for all and work activities

Table 8.8 summarises the sequences lengths of the baseline, deterministic, and the target approaches. The results indicate that the deterministic approach generates activities (all and work) that are longer than the M-IFN and shorter than activities generated by the target approach.

	Baseline	Deterministic	Target
All activities	3.3	3.45	3.94 (C4.5)
Work Activities	1.2	1.44	1.5 (C4.5)

Table 8.8 Mean Length of all and work activity sequences lengths

The validation at the activity pattern level further confirms that using a probabilistic action assignment rule for rule-based activity-based models enhances the predictive performance of models.

Work Activity Trip Matrix and Trips Start Time Level Accuracy Analysis (Spatial and Temporal Resolutions)

Table 8.9 shows the correlation coefficient between observed and predicted work trip OD matrices for the model using the deterministic action assignment rules. The correlation coefficient for the work trips OD matrices is calculated at 0.81 for the training set and 0.83 for the test set. While for the work activity start time per hour of the day the correlation coefficient reported quite high correlation coefficient with 0.93 and 0.91 for training and test sets respectively. The results obtained at this level are comparable to those obtained using the probabilistic action assignment rules.

	Work activity trip matrix level	Work activity Start time level
Training	0.81	0.94
Test	0.83	0.91

Table 0.9 Correlation Coefficients for work activity trips OD matrices and work activity start time per hour of the day

	Baseline	Deterministic	Target
OD Work activity trips	0.83	0.81	0.82
Work activity start time	0.82	0.91	0.85

Table 0.10 Spatial and temporal correlation coefficients

8.3 Decision Trees Classification methods parameters (pruning, minimum cases at leaf nodes)

All models in the FEATHERS / ALBATROSS models are trained by setting the minimum number of training cases at leaf nodes to 30. This number was obtained from original work done by the developers of ALBATROSS.

Nevertheless, it worked fine for the Dutch data. The analyses performed in previous chapters involved also training all decision tree models by setting the minimum number of cases at leaf nodes to 30. However, increasing or decreasing this number might result in better predictive performance, especially when using the Flemish data using ALBATROSS model.

In decision tree learning, the minimum number of cases at leaf nodes is a stopping criterion for decision tree growing. It is an important parameter, as it dictates the lowest number of training cases in a leaf node. If this number is low the resulting decision tree model is more specific to the training cases and this might lead to over-fitting. Over-fitting occurs when the induction algorithm generates a decision tree that perfectly fits the data in the training data set but lacks the capability of generalization of instances not present in the training set (Witten and Frank, 2005). Over-fitting is considered a problem in decision tree learning, because it results generating large rule sets and/or rules with very low predictive accuracy for unseen data (Witten and Frank, 2005). One solution to this problem is pruning, and so all decision tree models used in the analyses are trained with the pruning option turned on. Moreover, increasing the minimum number of cases at leaf nodes results in generating more generalized decision tree models and hence, helps to avoid over-fitting. However, there is no preferred method to achieve the best value for this parameter. For the purpose of the analysis in this chapter the minimum number of cases at leaf nodes will be experimented. The models are trained by increasing/decreasing this number to more/less than 30 cases and then investigate the resulting models. Furthermore, validate the models against test sets in an attempt to assess how sensitive the work activity process model to this parameter.

At first the models are trained by increasing the minimum number of instances at leaf nodes to 40. For decision steps (Work model), this results in under-fitting as the resulting model is similar to the M-IFN model explained in Chapter 6, with a decision tree model of two levels. And the validation results presented in Chapter 6 suggests that the C4.5 model with 30 as the minimum number of cases outperformed the M-IFN. Thus, increasing this number for this model is not

expected to enhance the performance. Considering the model at decision step 3 (More_Work_Ep) when training the model with a minimum number of cases at 40, results in a decision tree model with only one node, which always predict 0 class (one work episode). This model is similar to the deterministic action assignment rules model discussed above. Hence, increasing this number for this model will not increase the predictive performance as well.

Given that increasing the minimum number of case results in models that are not expected to increase the performance, the models are trained by decreasing the number of cases to 20 and 5 cases. For ease of reference the models trained with a minimum number of cases of 30, 20, and 5 cases will be referred to as *C4.5 M-30* (model generated in Chapter 6), *C4.5 M-20* and *C4.5 M-5* respectively. In the next subsections the validation of both models are discussed in more details.

Classifier level validation

At individual classifier level, considering the *Work* model at decision step 1 (Table 8.11) it is shown that both models reported approximately similar performance with a slight increase in favour of the C4.5 M-20 model. It is also noteworthy that drop in the accuracy in the test set is not significant especially in the C4.5 M-20. And this implies that generating a more generalized decision tree model, enhance model's performance. In addition, comparing the validation results with those obtained for the C4.5 M-30 model is also comparable with the C4.5 M-20 model. As the F-Measure and Brier Score for the C4.5 M-30 reported 0.60 and 0.115 respectively for the training set and an F-measure of 0.55 and 0.115 Brier Score for the test set.

With regard to the More_Work_Ep model at decision step 3 (Table 8.12), one can note that again both models obtained comparable results with a slight increase in performance for the C4.5 M-20 model. However, the C4.5 M-30 model, obtained in Chapter 7 (Tables 7.1 & 7.2), outperformed both the C4.5 M-

20 and C4.5 M-5 models. Hence, for such highly unbalanced data set it seems 30 is the preferred number of minimum cases at leaf nodes.

Decision step - 1 Work	Training set			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
C4.5 M-20	0.111059	0.602726	0.839053	0.615614
C4.5 M-5	0.113725	0.609533	0.848126	0.610152
Decision step - 1 Work	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
C4.5 M-20	0.116325	0.562753	0.833974	0.577362
C4.5 M-5	0.122238	0.557734	0.843966	0.551724

Table 8.11 C4.5 M-20 and M-5 Discrete choice classifiers accuracy statistics for the Work model

Decision step - 3 More_Work_Ep	Training set			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
C4.5 M-20	0.0825478	0.155738	0.879813	0.152
C4.5 M-5	0.108202	0.147368	0.905484	0.125561
Decision step - 3 More_Work_Ep	**Test set**			
Model	**Brier Score**	**Sensitivity**	**Specificity**	**F-Measure**
C4.5 M-20	0.111752	0.152542	0.872774	0.132353
C4.5 M-5	0.134237	0.169811	0.888041	0.138462

Table 0.12 C4.5 M-20 and M-5 Discrete choice classifiers accuracy statistics for the More_Work_Ep model

The baseline and target approaches accuracies (Table 8.13) for decision step 1 shows that they still serve as the lower and upper performance boundaries. Similarly for decision step 3 (Table 8.14) the accuracy measures, specifically the sensitivity and the F-Measure suggest that the M-IFN approach is the baseline performance model and the target approach outperforms the C4.5 M-5 and C4.5 M-20 models.

Approach	1-Brier Score	Sensitivity	Specificity	F-Measure
Baseline	0.72	0.30	0.85	0.22
C4.5 M-5	0.88	0.55	0.83	0.55
C4.5 M-20	0.88	0.56	0.84	0.58
Target	0.88	0.81	0.84	0.73

Table 0.13 Baseline, deterministic-C4.5, and target accuracy statistics for decision step 1

Model	1-Brier Score	Sensitivity	Specificity	F-Measure
Base line	0.86	0.12	0.85	0.11
C4.5 M-5	0.87	0.17	0.88	0.13
C4.5 M-20	0.89	0.15	0.87	0.14
Target	0.87	0.19	0.90	0.23

Table 8.14 Baseline, deterministic-C4.5, and target accuracy statistics for decision step 3

Activity pattern level

Tables 8.15 and 8.16 show the confusion matrices for work activity sequences lengths. The results show that both models (C4.5 M-20 and C4.5 M-5) approximately obtained similar results, with the C4.5 M-5 model being more accurate in predicting work sequences with two and four symbols.

	Training set	Predicted					Test set	Predicted			
C4.5 M-20	Sequence Length	1	2	4	Total	C4.5 M-20	Sequence Length	1	2	4	Total
Observed	1	0.78	0.19	0.03	0.52	Observed	1	0.80	0.17	0.03	0.52
	2	0.45	0.50	0.05	0.36		2	0.35	0.57	0.07	0.32
	4	0.46	0.50	0.04	0.12		4	0.46	0.52	0.02	0.16
	Total	0.63	0.34	0.04	589.00		Total	0.60	0.36	0.04	298.00

Table 8.15 C4.5 M-20 confusion matrix for work activity sequences length

	Training set	Predicted					Test set	Predicted			
C4.5 M-5	Sequence Length	1	2	4	Total	C4.5 M-5	Sequence Length	1	2	4	Total
Observed	1	0.79	0.19	0.02	0.52	Observed	1	0.82	0.14	0.04	0.52
	2	0.42	0.51	0.06	0.36		2	0.35	0.55	0.09	0.32
	4	0.41	0.53	0.06	0.12		4	0.35	0.54	0.10	0.16
	Total	0.61	0.35	0.04	589.00		Total	0.59	0.34	0.07	298.00

Table 8.16 C4.5 M-5 confusion matrix for work activity sequences length

At the Activity pattern level (Tables 8.17 and 8.18) the SAM distances for all activities and work activities are calculated. The results suggest that the C4.5 M-20 model outperformed the C4.5 M-5 model. The results at this level conform to those obtained at the individual classifier level. However, the C4.5 M-20 model reported smaller SAM distance than both models. The C4.5 M-20 reported a SAM distances at 4.53 and 4.77 for Training and test sets respectively for all activities, while for work activities a SAM distance at 0.59 for training set and 0.65 for the test set.

M-5	All Activities	Work Activities
Training	4.66	0.61
Test	4.96	0.69

Table 8.17 SAM distance for C4.5 M-5 activity pattern

M-20	All Activities	Work Activities
Training	4.53	0.59
Test	4.77	0.65

Table 0.18 SAM distance for C4.5 M-20 activity pattern

Comparing the C4.5 M-20 and M-5 models with the baseline and target approaches, the models generated shorter activity sequences than the target approach and longer activity sequences than the M-IFN approach (Table 8.19).

	Baseline	C4.5 M-5	C4.5 M-20	Target
All activities	3.3	3.17	3.4	3.94 (C4.5)
Work Activities	1.2	1.5	1.5	1.5 (C4.5)

Table 0.19 Mean Length of all and work activity sequences lengths (C4.5 M-20 and M-5)

Work Activity Trip Matrix and Trips Start Time Level Accuracy Analysis (Spatial and Temporal Resolutions)

In Table 8.20 the training and test sets correlation coefficient between observed and predicted work trips OD matrices for the C4.5 M-20 and C4.5 M-5 models is reported. As reported at previous validation results both models reported about similar performance. On the other hand, the C4.5 M-30 model reported higher correlation coefficient than both models.

The validation at the temporal resolution (the work activity start time per hour of the day) is shown in Table 8.21. Both models obtained similar performance.

	Work activity trip matrix level	
Dataset	**C4.5 M-20**	**C4.5 M-5**
Training	0.81	0.80
Test	0.79	0.79

Table 8.20 C4.5 M-20 and C4.5 M-5 Work activity trip matrix correlation coefficients

	Work activity Start time level	
Dataset	**C4.5 M-20**	**C4.5 M-5**
Training	0.92	0.91
Test	0.92	0.92

Table 8.21 C4.5 M-20 and C4.5 M-5 Work activity trip matrix correlation coefficients

As shown in Table 8.22 the correlation coefficients are comparable. As shown in previous analysis the validation at this level cannot detect local changes.

	Baseline	C4.5 M-5	C4.5 M-20	Target
OD Work activity trips	0.83	0.79	0.79	0.82
Work activity start time	0.82	0.92	0.92	0.85

Table 0.22 Spatial and temporal correlation coefficients

8.4 Conclusion and Discussion

In this chapter a sensitivity analysis on individual classifiers' action is conducted. First by experimenting the effect of replacing the probabilistic action assignment rules with a deterministic action assignment rules used for predicting values of cases under study. Second by assessing the performance of decision tree models by attempting to fine tune the minimum number of cases at leaf nodes. The experimental results reveal that adapting a deterministic action assignment rule for the decision tree models trained with the Flemish data did not increase the overall performance of the model. Although for some models (the Work model at decision step 1) the performance using the deterministic action assignment rules has increased. Hence, considering the highly unbalanced nature of some data sets (More_Work_Ep at decision step3) resulted in an

insensible model. Therefore, for the Flemish data adapting probabilistic action assignment rules, as already adapted in the original ALBATROSS model for the Dutch data, is still the best approach.

The results of the sensitivity analysis of the decision tree parameters suggested that increasing the minimum number of cases at leaf nodes will result in model under-fitting and hence, the predictive performance is decreased. Therefore, the models are trained by decreasing the minimum number of instances at leaf nodes to 20 and 5. The experimental results show that by decreasing the number to 20 cases at leaf nodes has no significant effect on the predictive performance of the model. In addition, decreasing the number to 5 results in increase over-fitting and thus, decreases the predictive performance of the models.

References

Arentze, T. A., and H. J. P. Timmermans (2005). Albatross 2: A Learning-Based Transportation Oriented Simulation System. European Institute of Retailing and Services Studies, Eindhoven, Netherlands.

Arentze, T.A., and H.J.P. Timmermans (2003) Measuring the Goodness-of-Fit of Decision-Tree Models of Discrete and Continuous Activity-Travel Choice: Methods and Empirical Illustration, *Journal of Geographical systems*, **4**, 1-22.

Bellemans T., Janssens D.., Wets G., Arentze, T., and Timmermans H. (2010). Implementation Framework and Development Trajectory of the FEATHERS Activity-Based Simulation Platform. Proceedings of the Annual Meeting of the Transportation Research Board

Kass, G.V. (1980) An exploratory technique for investigating large quantities of categorical data. Applied Statistics, 29, 119{127.

Lim, T.S., Loh, W.Y. and Shih, Y.S. (2000), "A Comparison of Prediction Accuracy, Complexity, and Training Time for Thirty-three Old and New Classification Algorithms" Machine Learning, 40, 203-228.

Quinlan, J.R. (1986), Induction of Decision Trees, Machine Learning, vol. 11, no. 1, pp. 81–106.

Rasouli, S., T.A Arentze and H.J.P. Timmermans (2011), Error propagation in complex large-scale computational process models of activity-travel behavior. In: W.Y. Szeto, S.C. Wong and N.N. Sze (ed.), Transportdynamics, HKSTS, Hong Kong, China, pp. 291-298.

Witten, I. H. and Frank, E. (2005). Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann, second edition.

Chapter 9

Final Discussion and Conclusions

9.1 Introduction

The current version of ALBATROSS employs 26 decision steps that are necessary to predict activity schedules for each person under study. In addition, in the current version of ALBATROSS, the scheduling process model contains two interesting features, the activation dependency feature and the attributes interdependency feature. The activation dependency feature affects the execution path of the process model depending on decision models (steps) outcome. While the attributes interdependency feature suggests adding decision outcomes of decision steps as attributes in the data set of subsequent decision steps. These two features and other factors such as, the quality of the data that are used to train the models, the classification method that is used at decision models, and the data representation inside the process mode, are important factors that influence the predictive performance of the process model.

This thesis discussed three contributions related to computational process activity-based models in the context of the ALBATROSS model. The first contribution was to examine the factors that improve the predictive performance of the scheduling process models integrated into the FEATALB framework. This goal was achieved by training the decision models in three different approaches. First by modeling all the decision models in the process model simultaneously, using a multi-target classification method. Using a multi-target classification method eliminates the activation dependency and attributes interdependencies features and it has the lowest fitting capacity. Second by training the decision models without the attributes interdependencies. This allowed investigating the added value of this feature in the model. Third by training the models at decision steps preserving the attributes interdependencies among models (fully-informed

approach) while including observed rather predicted decision outcomes in subsequent decision steps. To investigate the classification method factor, the non-informed and fully-informed approaches are examined using three classification methods, CHAID, C4.5, and Logistic regression methods.
The second contribution was related to investigating the data representation factor to improve the predictive performance of process models used in activity-based models. This was achieved by presenting three different process models, i.e. the activation dependency feature by changing the order of decision models in the process model.

The third contribution was related to studying the sensitivity of the models at each decision step (decision tree models). The sensitivity analysis was performed by experimenting two important factors used in the decision tree models in FEATALB. The first sensitivity factor involved identifying the ideal number of minimum cases per leaf node while training the decision tree models. In ALBATROSS this number was set to 30, however, this number was set to be used in the Dutch data. For the Flemish data, a different number might improve the performance. The second sensitivity factor is the action assignment rule used in predicting values at decision steps. ALBATROSS suggests a probabilistic action assignment rule which considers the probability of predicting a specific class. Rather than predicting a class variable according to the plurality rule. The work reported in this thesis was conducted in within the FEATALB framework. The FEATALB framework is based on the FEATHERS framework, which currently integrates the ALBATROSS model as its core scheduling system.

9.2 The FEATALB framework

The FEATALB framework is based on the FEATHERS framework. The FEATHERS framework is developed to facilitate the development of a modular activity-based model for transportation demand in Flanders (Belgium). At first the framework adopted a four-stage development trajectory, for a smooth transition from the four-step models towards static activity-based models in the short term

and dynamic activity-based models in the longer. In this study, Flanders (Belgium) is used as the study area.

To include ALBATROSS in FEATHERS, the model parameters were modified to fit the Flemish data. The ALBATROSS model and its components have been studied in details. However, some practical limitations were determined that restrained further experimentations and there was a need for new implementations. Some parts of the model were re-implemented. The implementation involved using technologies to boost the design of experiments conducted in this thesis.

New functionalities were added. First, in ALBATROSS, at each decision step in the process model is controlled by a CHAID decision tree. The process model contains 26 decision trees which are hard coded in the system. This approach makes experimenting with other induction or classification methods inapplicable. Therefore, the platform was extended to employ other induction methods. The extension involved the implementation of a library (*DecisionMaker*) that deploys data mining classification methods. The additional functionality allowed us to train models outside FEATALB, using data mining packages that can export Predictive Model Markup Language (PMML).

Second, a database was incorporated with the system to capture the output (predicted) schedules of persons. This functionality allowed to generate different generate several statistics and ODs with many dimensions from the database version of the schedule without needing to run the simulation each time. Third, the validation of the models at the individual classifier, the activity pattern, and the spatial and temporal levels are also implemented. The models at the individual classifier level are validated using Confusion Matrix statistics and the Brier Score. Where at the activity pattern level, the models are validated using the Sequence Alignment Method (SAM). And at the spatial temporal level, OD matrices and other statistics are calculated.

9.3 Predictive performance

The predictive performance of the approaches and models experimented was assessed at three levels. The models at the individual classifier level are validated

using Confusion Matrix statistics and the Brier Score. Where at the activity pattern level, the models are validated using the Sequence Alignment Method (SAM). And at the spatial and temporal level, OD matrices and work activity start times statistics are calculated using the correlation coefficient.

Let us consider the predictive performance (or the predictive accuracy) boundaries of the models at each decision model (individual classifier) level. The accuracy measure selected is the F-Measure, since it the weighted average of the sensitivity and precision of a classifier. Sensitivity approximates the probability of the positive class being correctly classified. While precision or positive predictive value is defined as the proportion of the true positives against all the positive results. The F-measure focuses more on the dropout class. An F-measure value reaches its best value at 1 and its worst value at 0.
Figure 9.1 shows the performance for discrete choice models i.e. the *Work* (decision step 1) and *Nep* models, for all the settings and approaches used the thesis.

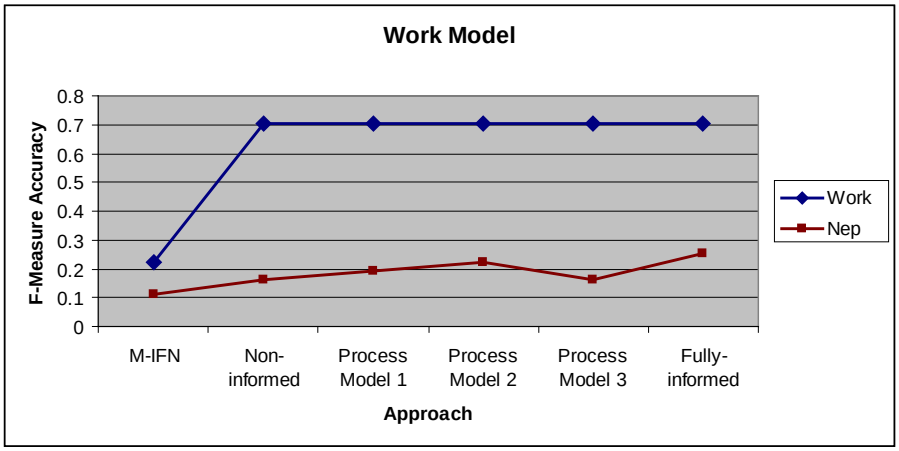


Figure 0.1 Test set performance of discrete choice models for all approaches

Figure 9.1 shows that the M-IFN method clearly sets the lowest performance bound below which all other approaches perform, for both decision models. Consequently the fully-informed approach sets the highest performance bound

above all approaches. The *work* model reported the same performance for all approaches except the M-IFN because it is the first decision model and thus, no additional attributes are added. For the *Nep* model in the non-informed approach and the model used in process model 3 obtained similar performance because the *Nep* model's order in this process model is the second and so no additional attributes are added. The model in process model 2 achieved better accuracy below the model in the fully-informed approach.

Figure 9.2 show the RAE of continuous models for all approaches. The results show again that M-IFN approach reported the highest RAE and the fully-informed approach reported the Lowest. This confirms the performance bounds set by these two approaches.

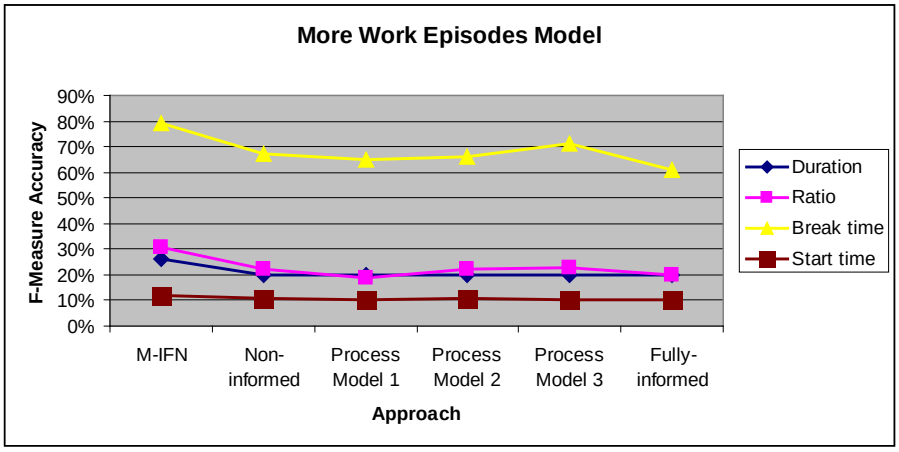


Figure 0.2 Test set performance of continuous models for all approaches

The duration model's RAE was approximately the same because in process models 1 and 2 it was kept in the same order in the decision sequence. Nevertheless, in process model 3 it was moved downwards, which allowed extra attributes (decision outcomes) to be added. But the RAE did not decrease because the extra attributes were not relevant for the decision tree learning method.

The results at individual decision models show that the attribute interdependencies feature, which is adopted in the original ALBATROSS model,

is an important factor that enhances the predictive performance of rule-based activity-based models. Furthermore, as expressed in the literature, different decision tree models achieved similar performance. For highly unbalanced data sets Logit models did not compete, in terms of predictive behaviour with decision tree models.

Considering the aggregate validation levels (i.e. the sequence alignment measures (SAM) and the correlation coefficients of the OD matrices and work activity start times), apparently the results somewhat differ, especially at the spatial and temporal levels. The SAM measures determine the dissimilarity between the observed and predicted sequences of activities and should be as low as possible. Figure 9.3 show the SAM measure for all approaches. The M-IFN and the fully-informed approaches set the lower and higher performance bounds respectively.

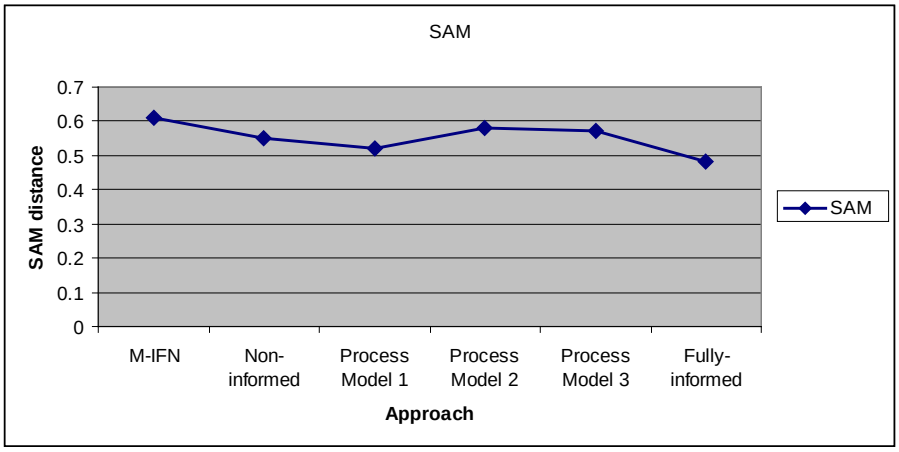


Figure 0.3 Test set SAM distance of work activities for all approaches

On the other hand, for other approaches, the non-informed approach reported lower SAM distance than process models 2 and 3. Therefore, it seemed that the at the activity pattern validation level, the SAM measure cannot detect local changes (changing the order of decision models i.e. activation dependency) in the process model.

The validation at the spatial and temporal dimensions, the results showed that at this aggregate level, the local changes are not detected. As shown in Figure 9.4 the correlation coefficient for the fully-informed is the highest (upper bound). However, the M-IFN approach obtained higher correlation coefficient between observed and predicted work activity ODs than the non-informed approach and the model produced by process model 1.

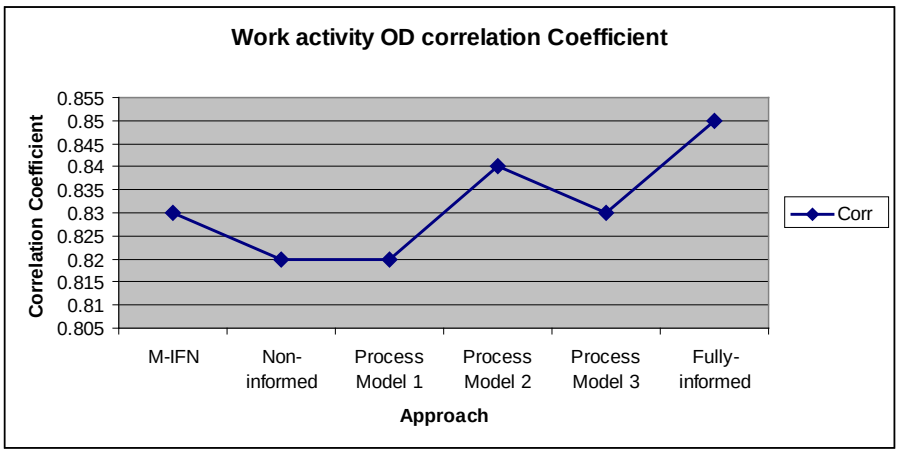


Figure 0.4 Test set correlation coefficient for work activity ODs

In conclusion the results of analyzing the factors affecting the predictive performance of activity-based models show that the attributes interdependencies feature is a critical factor. Maintaining this feature in activity-based models enhances the predictive performance. However, in this context, another factor that affects the predictive performance of process models is the relevance of the decision outcome that is added in subsequent decision models in predicting the class variable. On the other hand the results also suggest that the disposition of decision steps (activation dependency feature) or experiment other data representations within the work activity process model does not lead to significantly improve the predictive performance of the model. This is confirmed by the validation at the aggregated levels (activity pattern and Spatial and

temporal levels). Hence, using the currently implemented work activity process model can achieve satisfactory work activity schedules.

9.4 Model sensitivity

The action assignment rules, for discrete and continuous models, developed in ALBATROSS suggest probability distribution between classes. This probabilistic action assignment rule was developed by the authors to serve the purposes of generating bias free predictions and consider the effects of space-time constraints of activity-based models. In addition, decision tree models were trained setting the minimum number of leaf nodes to 30 cases. The same approaches are used in the FEATALB framework to predict schedules for the Flemish population. However, these approaches were developed and deployed for the Dutch data. To further confirm if they are also fit for the Flemish data, a deterministic action assignment rule was adapted. In addition, the decision tree models were trained setting the minimum number of cases at leaf nodes to 5 and 20 cases.

The results show that using a deterministic action assignment rule for decision trees decrease the predictive performance of activity-based models. Therefore, for the Flemish data adapting probabilistic action assignment rules, as already adapted in the original ALBATROSS model for the Dutch data, is still the best approach.

Considering the results of experiment the decision tree parameters (modifying the minimum number of cases at leaf nodes) suggested that increasing the minimum number of cases at leaf nodes to more than 30 cases will result in model under-fitting. And hence, the predictive performance is decreased. Therefore, the models are trained by decreasing the minimum number of cases at leaf nodes to 20 and 5. The experimental results show that by decreasing the number to 20 cases at leaf nodes has no significant effect on the predictive performance of the model. In addition, decreasing the number to 5 results in

increase over-fitting and thus, decreases the predictive performance of the models.

9.4 Future Research

Extending the FEATHERS framework, as an experimentation system to implement and deploy different classification methods accomplished in this PhD research was a first step towards further analyse the predictive performance of the scheduler process model. In addition, the ability to experiment the ALBATROSS system using classification methods other than CHAID was important to eliminate the black-box effect of the scheduler. However, to continue the research in this direction extra validation levels, and calculating descriptive statistics as the scheduler executes will provide more information. Hence, these methods need to be performed incrementally as the execution of the process model evolves, given the evolving nature of the scheduler. In addition implement more complex classification methods such as neural networks, Support Vector Machines (SVM) … etc.

The development of a new validation level other than the three validation levels used in this dissertation. The proposed validation level will be used as an evolving validation criterion given the evolving nature of the scheduler of the process model. Efforts already started to build such validation method using the SAM measure to measure how close are predicted decisions to the observed values incrementally as the execution of the process model evolves. This validation method mimics the scheduler activities as the prediction at decision steps is performed. Consequently, validating the models as the scheduling evolves will allow for further understand the process model and identify critical decision models that influence the predictive performance. Further be able to measure the inflation of errors as the process executes and how this affects the predictive performance of the process model. The inflation of errors is the fluctuation of taking a wrong decision at a specific decision step.

Analysing the predicative performance of rule-based process models using descriptive statistics at decision models as the scheduler evolves. Such descriptive statistics might be used specifically at continuous decision models, such as duration, ratio, break time and start time of work activities. These descriptive statistics will provide information to further understand the scheduler and identify critical decision steps. Furthermore, be able to determine how errors are inflated as the process model executes.

Furthermore, the predictive performance of the models presented in this research will further be validated at the new validation level to further confirm and performance bounds that were presented in chapter 6.