User profiling and classification for fraud detection in mobile communications networks

Jaakko Hollmén

Helsinki University of Technology
Department of Computer Science and Engineering
Laboratory of Computer and Information Science

Dissertation for the degree of Doctor of Science in Technology to be presented with due permission of the Department of Computer Science and Engineering, for public examination and debate in Auditorium T2 at Helsinki University of Technology (Espoo, Finland) on the 19th of December, 2000, at 12 noon.

Hollmén, J., **User profiling and classification for fraud detection in mobile communications networks**. ISBN 951-22-5239-2. (also published in print ISBN 951-666-555-1, ISSN 1456-9418). viii + 47 pp. UDC 004.032.26:519.21:621.391.

Keywords: fraud detection, telecommunication, neural networks, probabilistic models, data analysis, data mining.

ABSTRACT

The topic of this thesis is fraud detection in mobile communications networks by means of user profiling and classification techniques. The goal is to first identify relevant user groups based on call data and then to assign a user to a relevant group. Fraud may be defined as a dishonest or illegal use of services, with the intention to avoid service charges. Fraud detection is an important application, since network operators lose a relevant portion of their revenue to fraud. Whereas the intentions of the mobile phone users cannot be observed, it is assumed that the intentions are reflected in the call data. The call data is subsequently used in describing behavioral patterns of users. Neural networks and probabilistic models are employed in learning these usage patterns from call data. These models are used either to detect abrupt changes in established usage patterns or to recognize typical usage patterns of fraud. The methods are shown to be effective in detecting fraudulent behavior by empirically testing the methods with data from real mobile communications networks.

Preface

The work presented in this thesis was carried out at the Laboratory of Computer and Information Science in the Department of Computer Science and Engineering of the Helsinki University of Technology and Siemens Corporate Technology, Information and Communications, Department of Neural Computation in Munich, Germany. Funding was received from both institutions. Additional financial support was received from the Finnish Foundation for the Promotion of Technology, the Emil Aaltonen Foundation, and the NEuroNet — the European Network of Excellence in neural networks, set up under European Commission's ESPRIT III Program. A travel grant was received from the NIPS Foundation.

I wish to thank Professor Olli Simula for supervising my graduate studies and for guidance and encouragement during the thesis work. The Department of Neural Computation at Siemens Corporate Technology, headed by Professor Bernd Schürmann, provided an interesting subject for research. I would like to thank Michiaki Taniguchi and Volker Tresp, who lead the research groups I worked with at Siemens. I am especially indebted to Volker Tresp for guiding my research and for introducing me to probabilistic modeling. I also wish to thank my co-authors and colleagues for their collaboration. Academy Professor Erkki Oja and Academician Teuvo Kohonen have influenced this work through their pioneering work, also the Laboratory of Computer and Information Science and the Neural Networks Research Centre provided excellent facilities for research under their management. I wish to thank Professors Henry Tirri and Jyrki Joutsensalo for reviewing the manuscript of the thesis. I thank Docent Aapo Hyvärinen and Professor Mark Girolami for comments on the work. Finally, without the support and encouragement of my family, Suvi and Risto, this thesis work would not have been possible.

Contents

List of publications

This thesis consists of an introduction and the following publications:

Publication 1: Alhoniemi, E., J. Hollmén, O. Simula, and J. Vesanto (1999). Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering 6*(1), 3–14.

Publication 2: Taniguchi, M., M. Haft, J. Hollmén, and V. Tresp (1998). Fraud detection in communication networks using neural and probabilistic methods. In *Proceedings of the 1998 IEEE International Conference in Acoustics, Speech and Signal Processing (ICASSP'98)*, Volume II, pp. 1241–1244.

Publication 3: Hollmén, J. and V. Tresp (1999). Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. In M. Kearns, S. Solla, and D. Cohn (Eds.), *Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference (NIPS'11)*, pp. 889–895. MIT Press.

Publication 4: Hollmén, J., V. Tresp, and O. Simula (1999). A self-organizing map for clustering probabilistic models. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN'99)*, Volume 2, pp. 946–951. IEE.

Publication 5: Hollmén, J., M. Skubacz, and M. Taniguchi (2000). Input dependent misclassification costs for cost-sensitive classification. In N. Ebecken and C. Brebbia (Eds.), *DATA MINING II — Proceedings of the Second International Conference on Data Mining 2000*, pp. 495–503. WIT Press.

Publication 6: Hollmén, J. and V. Tresp (2000). A hidden markov model for metric and event-based data. In *Proceedings of EUSIPCO 2000 — X European Signal Processing Conference*, Volume II, pp. 737–740.

This numbering is used in the main text when referring to the publications.

Abbreviations and notations

BMU	Best-Matching Unit (also winner unit)
DAG	Directed Acyclic Graph
EM	Expectation Maximization algorithm
GSM	Global System for Mobile communications
HMM	Hidden Markov Model
KL	Kullback-Leibler distance
LVQ	Learning Vector Quantization
ROC	Receiver Operating Characteristic curve
SOM	Self-Organizing Map
$\alpha(t)$	adaptation gain value, also learning rate
c	index of the winner unit
$\delta(x - x_i)$	unit impulse function at x_i
$\partial/\partial x$	partial derivative with regard to x
i, k	unit index
$h^c(t, k)$	neighborhood kernel function
λ_{ij}	cost of classifying j as i
$\lambda_{ij}(x)$	cost of classifying j as i parameterized by data
$m^i(t), m^i$	weight vector of the unit i
$P(S)$	probability of hidden state vector $s_1, \ldots, s_T$
$P(s_t)$	probability of a hidden variable s at time t
$P(s_t\|s_{t-1})$	conditional probability of s_t given s_{t-1}
$p(x)$	probability density of x
$q(x;\ \theta)$	probability density of x (parameterized by θ)
r^k, r^c	location vector inside the array of neurons
$R(\alpha_i\|x)$	conditional risk of classifying x to the class i
$\sigma(t)$	neighborhood kernel width function
$x(t), x_i$	measurement vector
$x \sim p(x)$	x is distributed according to $p(x)$
Y	observed variable $y_1, \ldots, y_T$
y_t	observed variable y at time t
$\|.\|$	Euclidean distance

Chapter 1

Introduction

The topic of this thesis is fraud detection in mobile communications networks by means of user profiling and classification techniques. User profiling is the process of modeling characteristic aspects of user behavior. In user classification, users are assigned to distinctive groups.

Fraud may be defined as a dishonest or illegal use of services, with the intention to avoid service charges. With the aid of the fraud detection models, fraudulent activity in a mobile communications network may be revealed. This is beneficial to the network operator, who may lose several percent of revenue to fraud, since the service charges from the fraudulent activity remain uncollected. Apart from fraud detection, user profiling efforts in telecommunications may be further motivated by the need to understand the behavior of customers to enable provision of matching services and to improve operations.

Fraud is defined through the unobserved intentions of the mobile phone users. However, the intentions are reflected in the observed call data, which is subsequently used in describing behavioral patterns of users. The task is to use the call data to learn models of calling behavior so that these models make inferences about users' intentions. Neural networks and probabilistic models are employed in learning these usage patterns from call data. Learning in this context means adaptation of the parameterized models so that the inherent problem structure is coded in the model. Obviously, there is no specific sequence of calls that would be fraudulent with absolute certainty. In fact, the same sequence of calls could as well be fraudulent or normal. Therefore, uncertainty in modeling the problem is needed. This is naturally embodied in the framework of probabilistic models.

Two complementary approaches to fraud detection are used in this thesis. In the differential approach, a model of recent behavior is used in quantifying novelty found in the future call data so as to detect abrupt changes in the calling behavior, which may be a consequence of fraud. In the absolute approach, models typifying fraudulent and normal behavior are used to

determine the most likely mode.

Chapter 2 introduces the problem of fraud detection and presents a review of the published works in telecommunications fraud detection. Related fields such as intrusion detection in computer systems and credit card fraud detection are also briefly reviewed. In Chapter 3, the call data used in this thesis is described. Chapter 4 forms the core of this thesis, where the novel developments are put in a broader framework. The chapter starts by introducing probabilistic networks, a framework under which mixture models (**Publication 2**), regime-switching models (**Publication 3**), and extensions of hidden Markov models (**Publication 6**) are described. The chapter continues with the presentation of Self-Organizing Maps, which are applied in a related application in process monitoring (**Publication 1**) and in clustering probabilistic models (**Publication 4**). Since detection is inherently a discrimination problem, discriminative learning on top of the Self-Organizing Map is presented in the context of Learning Vector Quantization (**Publication 7**). The chapter proceeds by introducing cost models that can be used in expressing *user-specific* costs (**Publication 5**). Chapter 4 ends with a description of measuring the quality of the models and related discussion. The work is summarized in Chapter 5. Chapter 6 lists the contents of the publications and contributions of the author.

Chapter 2

Fraud detection

This chapter introduces the problem of fraud detection, starting from definitions and proceeding to a review of previous work. The end of the chapter discusses the related work in this area.

2.1 Introduction

In this section, fraud is defined and the development of fraud detection systems is motivated. Some historical background is used to motivate the user profiling approaches in fraud detection.

2.1.1 Definition of fraud

Many definitions in the literature exist, where the intention of the subscriber plays a central role. Johnson (1996) defines fraud as any transmission of voice or data across a telecommunications network where the intent of the sender is to avoid or reduce legitimate call charges. In similar vein, Davis and Goyal (1993) define fraud as obtaining unbillable services and undeserved fees. According to Johnson (1996), the serious fraudster sees himself as an entrepreneur, admittedly utilizing illegal methods, but motivated and directed by essentially the same issues of cost, marketing, pricing, network design and operations as any legitimate network operator. Hoath (1998) considers fraud as attractive from the fraudsters' point of view, since detection risk is low, no special equipment is needed, and the product in question is easily converted to cash. Although the term fraud has a particular meaning in legislation, this established term is used broadly to mean misuse, dishonest intention or improper conduct without implying any legal consequences.

2.1.2 Motivation for fraud detection

Following the definition of fraud, it is easy to state the losses caused by fraud as primary motivation for fraud detection. In fact, the telecommunications

industry suffers losses in the order of billions of US dollars annually due to fraud in its networks (Davis and Goyal 1993; Johnson 1996; Parker 1996; O'Shea 1997; Pequeno 1997; Hoath 1998). In addition to financial losses, fraud may cause distress, loss of service, and loss of customer confidence (Hoath 1998). The financial losses account for about 2 percent to 6 percent of the total revenue of network operators, thus playing a significant role in total earnings. However, as noted by Barson et al. (1996), it is difficult to provide precise estimates, since some fraud may be never detected, and the operators are reluctant to reveal figures on fraud losses. Since the operators are facing increasing competition and losses have been on the rise (Parker 1996), fraud has gone from being a problem carriers were willing to tolerate to being one that dominates the front pages of both trade and general press (O'Shea 1997). Johnson (1996) also affirms that network operators see call selling as a growing concern.

2.1.3 Development of fraud

Historically, earlier types of fraud used technological means to acquire free access. Cloning of mobile phones by creating copies of mobile terminals with identification numbers from legitimate subscribers was used as a means of gaining free access (Davis and Goyal 1993). In the era of analog mobile terminals, identification numbers could be easily captured by eavesdropping with suitable receiver equipment in public places, where mobile phones were evidently used. One specific type of fraud, tumbling, was quite prevalent in the United States (Davis and Goyal 1993). It exploited deficiencies in the validation of subscriber identity when a mobile phone subscription was used outside of the subscriber's home area. The fraudster kept tumbling (switching between) captured identification numbers to gain access. Davis and Goyal (1993) state that the tumbling and cloning fraud have been serious threats to operators' revenues. First fraud detection systems examined whether two instances of one subscription were used at the same time (overlapping calls detection mechanism) or at locations far apart in temporal proximity (velocity trap). Both the overlapping calls and the velocity trap try to detect the existence of two mobile phones with identical identification codes, clearly evidencing cloning. As a countermeasure to these fraud types, technological improvements were introduced.

However, new forms of fraud came into existence. A few years later, O'Shea (1997) reports the so-called subscription fraud to be the trendiest and the fastest-growing type of fraud. In similar spirit, Hoath (1998) characterizes subscription fraud as being probably the most significant and prevalent worldwide telecommunications fraud type. In subscription fraud, a fraudster obtains a subscription (possibly with false identification) and starts a fraudulent activity with no intention to pay the bill. It is indeed non-technical in nature and by call selling, the entrepreneur-minded fraud-

ster can generate significant revenues for a minimal investment in a very short period of time (Johnson 1996). From the above explanation it is evident that the detection mechanisms of the first generation soon became inadequate. The more advanced detection mechanisms must be based on the behavioral modeling of calling activity, which is also the subject of this thesis.

2.2 Previous work

In this section, published work with relevance to fraud detection in telecommunications networks is reviewed. Section 2.3 presents fraud detection methods in related fields, such as intrusion detection in computer systems, credit card fraud detection, and applications in other fields, such as health care fraud detection.

Fraud in telecommunications networks can be characterized by fraud scenarios, which essentially describe how the fraudster gained the illegitimate access to the network. Detection methodologies designed for one specific scenario are likely to miss plenty of the others. For example, velocity trap and overlapping calls detection methodologies are solely aimed at detecting cloned instances of mobile phones and do not catch any of the subscription fraud cases. As stated in Section 2.1.3, the nature of fraud has changed from cloning fraud to subscription fraud, which makes specialized detection methodologies inadequate. Instead, the focus is on the detection methodologies based on the calling activity (a stream of transactions), which in turn can be roughly divided into two categories. In *absolute analysis*, detection is based on the calling activity models of fraudulent behavior and normal behavior. *Differential analysis* approaches the problem of fraud detection by detecting sudden changes in behavior. Using differential analysis, methods typically alarm deviations from the established patterns of usage. When current behavior differs from the established model of behavior, alarm is raised. In both cases, the analysis methods are usually implemented by using probabilistic models, neural networks or rule-based systems. The two approaches are illustrated in Figure 2.1. In the following, some prominent work with relevance to the work presented in this thesis will be reviewed.

Davis and Goyal (1993) report on the use of a knowledge-based approach to analyze call records delivered from cellular switches in real time. They state that the application of uniform thresholds to all of a carrier's subscribers essentially forces comparison against a mythical average subscriber. Instead, they choose to model each subscriber individually and allow the subscribers' profile to be adaptive in time. In addition, they use knowledge about the general fraudulent behavior, for example, suspicious destination numbers. The analysis component in their system determines if the alarms, taken together, give enough evidence for the case to be reviewed by a hu-

Figure 2.1: Absolute analysis and differential analysis, the two main approaches to fraud detection, are illustrated using a probabilistic view. In absolute analysis, illustrated left, models of both normal (C_0) and fraudulent behavior (C_1) must be formulated. In differential analysis, one model is built assuming normal behavior (C_0) and any deviations from the established behavior are classified as fraudulent. The dashed lines indicate some arbitrary decision borders and the shaded area denotes the regions to be classified as fraudulent.

man analyst. In their conclusion, the system is credited with the ability to detect fraud quickly allowing the analysts to focus on the most likely and dangerous fraud cases.

In (Barson et al. 1996), the authors report their first experiments detecting fraud in a database of simulated calls. They use a supervised feedforward neural network to detect anomalous use. Six different user types are simulated stochastically according to the users' calling patterns. Two types of features are derived from this data, one set describing the recent use and the other set describing the longer-term behavior. Both are accumulated statistics of call data over time windows of different lengths. This data is used as input to the neural network. The performance of their classifier is estimated to be 92.5 % on the test data, which has limited value in the light of simulated data and the need to give class-specific estimates on accuracy. This work has also been reported in (Field and Hobson 1997).

Burge and Shawe-Taylor (1996, 1997) focus on unsupervised learning techniques in computing user profiles over sequences of call records. They apply their adaptive prototyping methods in creating models of recent and long-term behavior and calculate a distance measure between the two profiles. They discuss on-line estimation techniques as a solution to avoid storing call detail records for calculating statistics over a time period. Their user profiles are based on the user-specific prototypes, which model the probability distribution of the call starting times and call durations. A large change in user behavior profiles expressed by the Hellinger distance between profiles is reported as an alarm. In (Moreau and Vandewalle 1997; Moreau, Verrelst, and Vandewalle 1997), work on fraud detection based on supervised feedforward neural network techniques is reported. The authors criticize thresholding techniques by detecting excessive usage, since these might be the very best customers if these are legitimate users. In order to use supervised learning techniques, they manually label the estimated user profiles of longer

term and recent use, similar to those in (Burge and Shawe-Taylor 1997), into fraudulent and non-fraudulent and train their neural network on these user profiles. In (Moreau and Vandewalle 1997), they report having classified test data with detection probabilities in the range of 80 - 90 % and false alarm probabilities in the range of 2 - 5 %. Collaborative efforts of the two previous groups to develop a fraud detection system have been reported in (Moreau, Preenel, Burge, Shawe-Taylor, Störmann, and Cooke 1996; Burge, Shawe-Taylor, Moreau, Verrelst, Störmann, and Gosset 1997). Interesting in this context is the performance of the combination of the methods. In (Howard and Gosset 1998), performance of the combination of the tools is considered. They form an aggregated decision based on individual decisions of the rule-based tool, unsupervised and supervised user profiling tools with the help of logistic regression. They report improved results, particularly in the region of low false positives. In all, their combined tool detects 60 % of the fraudsters with a false alarm rate of 0.5 %.

Fawcett and Provost (1996, 1997) present rule-based methods for fraud detection. The authors use adaptive rule sets to uncover indicators of fraudulent behavior from a database of cellular calls. These indicators are used to create profiles, which then serve as features to a system that combines evidence from multiple profilers to generate alarms. They use rule selection to select a set of rules that span larger sets of fraudulent cases. Furthermore, these rules are used to formulate monitors, which are in turn pruned by a feature selection methodology. The output of these monitors is weighted together by a learning, linear threshold unit. They assess the results with a cost model in which misclassification cost is proportional to time.

Some work in fraud detection is based on detecting changes in geographical spread of call destinations under fraudulent activity. This view is promoted in (Yuhas 1993; Shortland and Scarfe 1994; Connor et al. 1995; Cox et al. 1997). Yuhas (1993) clusters call data for further visualization. Connor et al. (1995) in turn use neural networks in classification and some authors use human pattern recognition capabilities in recognizing fraud (Cox et al. 1997; Shortland and Scarfe 1994).

Fraud and uncollectible debt detection with Bayesian networks has been presented in (Ezawa 1995; Ezawa, Singh, and Norton 1996; Ezawa and Norton 1996). They perform variable and dependency selection on a Bayesian network. They also state that a Bayesian network that fits the database most accurately may be poor for a specific task such as classification. However, their problem formulation is to predict uncollectible debt, which includes cases where the intention was not fraudulent and which does not call for user profiles.

2.2.1 Comparisons of the published work

Comparisons between the approaches are difficult to make, since the performance assessment may differ, the difficulty of the problem varies and the problem is set up in different ways. Also, the available data from the domain may differ considerably. Fawcett and Provost (1999) state that because of the problem representations, it is difficult to compare different solutions. Collaborative work reported in (Moreau et al. 1996; Burge et al. 1997) is unique in the sense that they can combine results from several research groups based on the same framework for evaluation and the same data.

2.3 Related areas

Fawcett and Provost (1999) attempt to cast different fields, such as intrusion detection, fraud detection, network performance monitoring and news story monitoring into a common framework highlighting the similarities and differences. They introduce a problem class called *activity monitoring*, where the task is to detect the occurrence of interesting activity in a timely fashion based on the observations of entities in the population. On system level, monitoring of industrial processes has been earlier coined *process monitoring* and pursued by (Tryba and Goser 1991; Kasslin, Kangas, and Simula 1992; Simula, Alhoniemi, Hollmén, and Vesanto 1997; Alhoniemi, Hollmén, Simula, and Vesanto 1999; Simula, Ahola, Alhoniemi, Himberg, and Vesanto 1999). Process monitoring will be considered in more detail in Section 4.2.2. In the following, the focus is on the work done in intrusion detection in computer systems (Section 2.3.1), credit card fraud detection (Section 2.3.2), and fraud detection in other fields such as medical care and insurance (Section 2.3.3).

2.3.1 Intrusion detection on computer systems

The goal of intrusion detection is to discover unauthorized use of computer systems. Approaches to intrusion detection can be divided into two classes: anomaly detection and misuse detection. Anomaly detection, similarly to differential analysis, approaches the problem by attempting to find deviations from the established patterns of usage. Misuse detection, which in turn is similar to absolute analysis, compares the usage patterns to known techniques of compromising computer security (Kumar 1995). Architecturally, an intrusion detection may be based on audit data of a single host, or multiple hosts, or additionally on network traffic data. The earliest work on the subject is a study by Anderson (1980). An intrusion detection model by Denning (1987) is based on the hypothesis that security violations can be detected by monitoring a system's audit records for abnormal patterns of system usage. These early studies set the path for other work to follow.

Lunt (1990) considers combinations of anomaly detection and misuse detection to compensate for the shortcomings of each method. Overviews of intrusion detection methodologies can be found in (Lunt 1988; Lunt 1993; Frank 1994; Mukherjee, Heberlein, and Levitt 1994; Kumar 1995). A handbook on technical aspects of intrusion detection is found in (Northcutt 1999).

Neural networks have been used in intrusion detection. Fox, Henning, Reed, and Simonian (1990) use Self-Organizing Maps to identify anomalous system states to be post-processed by an expert system. Feed-forward neural networks have been used in (Tan 1995) to classify user behavior as normal or intrusive, and in (Ryan, Ling, and Miikkulainen 1997) to learn user profiles (prints) to recognize the legitimacy of the user.

Modeling the dynamic behavior of users is reported in (DuMouchel and Schonlau 1998). They model the user behavior with a transition matrix that models transition probabilities between subsequent commands of the user. Lane and Brodley (1997) present matching functions to compare current behavioral sequence to a historical profile to be used in intrusion detection. Other recent work addresses the problem of concept drift, changing tasks of legitimate computer users in intrusion detection (Lane and Brodley 1998).

Fawcett and Provost (1999) report work on transferring their fraud detection system to the intrusion detection domain. They report disappointing results, which means that, despite some similarities, transferability of the systems should not be taken for granted.

2.3.2 Credit card fraud detection

Credit card fraud detection aims at timely detection of credit card abuse. Dorronsoro et al. (1997) describe this domain as having two particular characteristics: a very limited time span for decisions and huge amount of credit card operations to be processed. Leonard (1993) sets forth an expert system model for detecting fraudulent usage of credit cards. Radial basis function neural networks have been used in the credit card fraud detection by Ghosh and Reilly (1994) and Hanagandi, Dhar, and Buescher (1996). In (Dorronsoro et al. 1997), an operational system for fraud detection of credit card operations based on a neural classifier is presented. Aleskerov et al. (1997) present a neural network based database mining system for credit card detection and test it on synthetically generated data. Stolfo et al. (1997) present a meta-learning approach in credit card fraud detection in order to combine results from multiple classifiers. Chan and Stolfo (1998) address the question of non-uniform class distributions in credit card fraud detection.

The problem of credit scoring does not share the same characteristics as the fraud detection in telecommunications and is not reviewed here. A survey of quantitative methods in credit management in a broader sense can be found in (Rosenberg and Gleit 1994).

2.3.3 Other work on fraud detection

There are numerous fields where one is interested in finding anomalous or illegitimate behavior based on the observed transactions. Similar work may be found in diverse fields, such as in insurance industry, health care, finance, and management.

Glasgow (1997) discusses risk in the insurance industry and divides it to two parts: risk as an essential element of the related underwriting task and the fraud risk. In health care fraud detection, knowledge-based systems have been applied in (Sokol 1998; Major and Riedinger 1992). He, Wang, Graco, and Hawkins (1997) present medical fraud detection by grouping practice profiles of medical doctors to normal and abnormal profiles with the aid of neural networks. An assessment of artificial intelligence technologies for detection of money laundering is presented in (Jensen 1997). Schuerman (1997) discusses risk management in the financial industry, and Barney (1995) deals with closely related trading fraud. Allen et al. (1996) transform financial transaction data to be visualized for further inspection by a domain expert. Management directed fraud has been examined by Menkus (1998) and by Curet, Jackson, and Tarar (1996). Fanning, Cogger, and Srivastava (1995) use neural networks in detecting management fraud.

2.4 Discussion

Fraud detection is usually approached by absolute or by differential analysis. Variations on the theme are due to the representation of the problem, the choice of model classes, the degree of available knowledge about known fraud scenarios, and the kind of available data exemplifying fraudulent and normal behavior.

Surprisingly, very little work exists on dynamic modeling of behavior, although many authors state fraud to be a dynamic phenomenon. Fawcett and Provost (1997), for example, doubt the usefulness of hidden Markov models in fraud detection as, in this domain, one is concerned with two states of nature and one single transition between them. In this thesis, these models are used extensively in temporal modeling of behavior. The dynamical modeling of behavioral patterns for fraud detection is one of the main contributions of the thesis.

The concept of learning has a central part in the thesis, and the models are implemented using neural networks and probabilistic models. The methods presented in this thesis solve the learning problem with a mixture setting of data. In this setting, one has access to data from normal accounts and accounts that *contain* fraudulent data. Learning from partially labeled data (as will be explained in Chapter 3) is a major advantage that saves the human labor needed in an extensive labeling effort.

Whereas the main topic of this thesis is fraud detection, the presented

methods in user profiling and classification have wider applicability. Interesting applications may be found in identifying user profiles in hypertext document navigation patterns or buying habits, for example.

Chapter 3

Call data

In this thesis, fraud detection is based on the calling activity of mobile phone subscribers. As mentioned earlier, the problem of fraud detection is to discover dishonest intention of the subscriber, which clearly can not be directly observed. Acknowledging that the intentions of the mobile phone subscribers are reflected in the calling behavior and thus in the observed calling data, the use of call data as a basis for modeling is well justified.

Conventionally, the calling activity is recorded for the purpose of billing in call records, which store attributes of calls, like the identity of the subscriber (IMSI, International Mobile Subscriber Identity), time of the call, duration of the call to mention a few. In all, dozens of attributes are stored for each call. In the context of GSM networks, the standard about administration of subscriber related events and call data in a digital cellular telecommunications system can be found in (European Telecommunications Standards Institute 1998).

3.1 Data Collection

In order to develop models of normal and fraudulent behavior and to be able to assess the diagnostic accuracy of the models, call data exhibiting both kinds of behavior is needed. Gathering normal call data is relatively easy as this mode dominates the population, but collecting fraudulent call data is more problematic. Fraudulent call data is relatively rare and the data collection involving human labor is expensive. In addition, the processing and storing of data is subject to restrictions due to legislation on privacy of data.

Procedures in data collecting differ both in the way they are conducted and in the way the data is grouped in the normal and fraudulent modes. In Sections 3.1.1 and 3.1.2 two ways of collecting fraud data for development of a fraud detection system are described.

3.1.1 Block crediting

After each billing period, telephone bills are calculated from the subscriber specific call data using appropriate tariffs (pricing) for each service. A bill is sent to the customer, who either approves or disapproves the billed amount. If a fraudster has exploited an account during the billing period, the customer is likely to disapprove the high cost of calling.

Fawcett and Provost (1997) describe the process of *block crediting*, where a representative of the operator and the defrauded customer together establish the range of dates during which the fraud occurred, and the calls within the range are credited to the customer. This effort involves a lot of human labor and is naturally expensive, and admittedly such a process is likely to contain errors. As a result, however, each call is labeled to legitimate or fraudulent class, which can be considered a relatively accurate labeling of data.

3.1.2 Velocity trap

It would be beneficial if a fraud detection system could be designed using data from normal and fraudulent accounts without extensive labeling involving human labor. One approach is to filter fraudulent call data from a large database by formulating an elementary fraud model and testing whether call data is fraudulent. This works under the assumption of cloning fraud and using a velocity trap as an elementary fraud model. Velocity trap alarms if calls are made from locations geographically far apart in temporal proximity. In essence, this sets a limit on the velocity a mobile phone subscriber may travel, hence the name.

Fraud data used in this thesis is filtered from a database of call data using a velocity trap detection mechanism. An important consequence of this is that the data does not contain information on which calls were fraudulent or which periods contained fraudulent activity. Data labeled as fraudulent is a sample from a *mixture* of normal and fraudulent data, the mixing coefficients being unknown and changing in time. The setting of data is illustrated in Figure 3.1. Therefore, call data is labeled to classes fraud and normal on a subscriber basis. No geographical information about the calls was available in the call data nor when the velocity trap gave an alarm. The database of fraudulent behavior contained call data of 304 subscribers during a period of 92 days. The normal call data spanned a period of 49 days and was assumed to contain no fraudulent activity. The number of users used in the publications was limited by the available resources. The use of the data may vary in the publications. Consult individual publications for details.

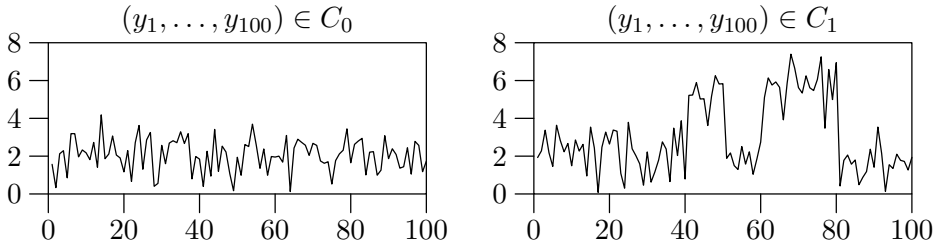

Figure 3.1: Examples of mixture data. In the left panel, the data belongs to class C_0, which has a Gaussian distribution $\mu_1 = 2, \sigma_1^2 = 1$. In the right panel, the data is labeled to belong to the class C_1, since it *contains* data with a Gaussian distribution $\mu_2 = 5, \sigma_2^2 = 1$, representative of class C_1. The data from this density is located at $t = 40, \ldots, 50$ and $t = 60, \ldots, 80$. However, these regions are not known in the data. This generated data illustrates the partially labeled call data used in this thesis: normal users are always normal, but fraudulent users behave sometimes normally and sometimes in a fraudulent fashion.

3.2 Representation of call data

Call records constituting the call data are transactions (or events) ordered in time. Each of the call records has a set of call attributes as described earlier. These attributes need to be converted to a form that is compatible with the model used. This conversion can take many forms. Three different data representations are used in this thesis.

3.2.1 Features through aggregation in time

In pattern recognition applications, the usual way to create input data for the model is through feature extraction. In feature extraction, descriptors or statistics of the domain are calculated from raw data. Usually, this process involves some form of aggregation.

In **Publication 2** (Taniguchi, Haft, Hollmén, and Tresp 1998), the detection is based on feature variables derived from call data. The unit of aggregation in time is one day. The feature mapping transforms the transaction data ordered in time to static variables residing in feature space. The features used reflect the daily usage of an account. Number of calls and summed length of calls to describe the daily usage of a mobile phone were used. National and international calls were regarded as different categories. Calls made during business hours, evening hours and night hours were also separated to sub-categories.

3.2.2 Dynamic description of call data

There is a connection between the length of the aggregation period used in feature extraction and the richness of description. It is interesting to consider representations that describe the instantaneous behavior of mobile phone subscribers by pushing the length of the aggregation period to the minimum at the price of a representational richness. In **Publication 3** (Hollmén and Tresp 1999), this kind of representation is used. The call data is sampled for one minute intervals, and the data indicates whether a mobile phone is used during a particular minute. The data for the minute t is then represented with $y_t \in \{0, 1\}$.

This representation describes the instantaneous calling behavior of mobile phone subscribers and permits the dynamic modeling of the calling behavior expressed with transitions from one time step to another. This representation is also the basis of modeling in **Publication 4** (Hollmén, Tresp, and Simula 1999) and **Publication 7** (Hollmén, Tresp, and Simula 2000).

3.2.3 Switching representation

In essence, the feature variables mediate information from the domain to the model used. Sometimes, the model class is limited to certain representations, like categorical data or metric data. In order to avoid compromising how the domain is described, models may be extended to handle data with a more unconventional representation.

The issue of changing representations between metric data and event-based data is reported in **Publication 6** (Hollmén and Tresp 2000). The fact that the data is switching between the continuous and the categorical representations is an artifact of a feature extraction process. When a faithful mapping of the domain is sought for, like in the user profiling problem, extending the model class becomes necessary. The extension is presented in the case of a hidden Markov model (Baum 1972; Juang and Rabiner 1991; Bengio 1999).

Chapter 4

User profiling and classification

This chapter introduces the user profiling and classification methods used for fraud detection. The purpose is to place the novelties found in the publications in a broader framework. The work on finite mixture models, hidden Markov models and hierarchical regime-switching models can be nicely described in the framework of probabilistic networks and therefore the concepts are presented on a general level. Learning in the maximum likelihood framework with the EM algorithm is also presented. In the sequel, Self-Organizing Maps and Learning Vector Quantization are presented with the appropriate extensions. The chapter proceeds with cost-sensitive classification methods and technical assessment methods for the fraud detection domain. The chapter ends with a discussion on the presented methods for fraud detection articulating their advantages and disadvantages.

4.1 Probabilistic networks

Probabilistic networks allow an efficient description of multivariate probability densities (Cowell, Dawid, Lauritzen, and Spiegelhalter 1999). Probabilistic formulations allow quantifying uncertainty in the conclusions made about the problem, which makes the framework of probabilistic networks appealing for real-world problems. Of particular interest here are the Bayesian networks (Cowell et al. 1999; Jensen 1996), which can be represented as directed acyclic graphs (DAG). A Bayesian network may be represented as a graph $\mathcal{G} = (V, E)$, where V is the set of vertices or nodes and E is the set of arcs, which is defined as an ordered set of vertices $E \subset V \times V$. The nodes of the graph correspond to the domain variables and an arc to the qualitative dependency between two variables (see Figure 4.1).

Graphical representation makes it easy to understand and manipulate networks. The term *graphical model* refers to this dual representation of

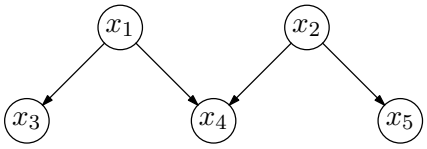

Figure 4.1: A simple Bayesian network is shown. Variables are marked with graph nodes, the dependency relationships as arcs. The joint probability density can be factorized as $P(x_1, x_2, x_3, x_4, x_5) = P(x_1)P(x_2)P(x_3|x_1)P(x_4|x_2, x_1)P(x_5|x_2)$.

probabilistic models as graphs. In the following sections, the concept of conditional independence is briefly described. It is used in defining qualitative relationships between the variables, whereas the distributional assumptions define the quantitative aspect of the probabilistic networks. Learning from data is then briefly described within the framework of maximum likelihood using the EM algorithm (Dempster, Laird, and Rubin 1977; McLahlan 1996).

4.1.1 Conditional independence

A problem domain consists of a set of random variables. A random variable is an unknown quantity that can take on one of a set of mutually exclusive and exhaustive outcomes (Cowell et al. 1999). The joint probability density $P(x_1, \ldots, x_n)$ of the random variables $x_1, \ldots, x_n$ can be decomposed according to the chain rule of probability (Equation 4.1) as

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|x_{i-1}, \ldots, x_1). \tag{4.1}$$

Each term in this factorization is a probability of a variable given all lower numbered variables. In real life, however, not all factors influence the others in a given domain, thus this kind of qualitative knowledge can be formulated by assuming conditional independence relations between the domain variables. The use of conditional independence assumptions allows one to construct global joint distribution from a set of local conditional probability distributions. Defining $\pi_i \subseteq \{x_1, \ldots, x_{i-1}\}$ as the parent set of x_i or the set of variables that renders x_i and $\{x_1, \ldots, x_{i-1}\}$ conditionally independent, the joint probability density can be written as

$$P(x_1, \ldots, x_n) = \prod_{i=1}^{n} P(x_i|\pi_i). \tag{4.2}$$

A Bayesian network defines this joint probability density as the product of local, conditional densities. The main contribution of the conditional

independence assumptions is that the expression for the joint probability density in Equation 4.2 is simpler than the trivial decomposition achieved by the application of chain rule of probability in Equation 4.1.

4.1.2 Distributional assumptions

Conditional independence assertions provide qualitative assumptions between variables in the probabilistic network. To further quantify these established relationships, one needs to define for every variable in the network the conditional probability distribution of the variable given its parents. Using classic estimation theory (Cherkassky and Mulier 1998), the probability distributions are specified to come from a parameterized family of distributions. In estimation, the parameters are determined so that the distribution approximates the distribution of the data.

If the observations in each component of a finite mixture model are distributed according to a Gaussian (normal) distribution, it is called the Gaussian mixture model (Redner and Walker 1984; Bishop 1996). In this thesis, this kind of model was used in **Publication 2** (Taniguchi, Haft, Hollmén, and Tresp 1998) to model the probability density of recent calling behavior to be used in novelty detection to detect changes in behavioral patterns. Discrete states in the models are best modeled with the assumption of multinomial distributions, in which the variable can be in one of many states of the variable. Exponential distribution was used for modeling call lengths in **Publication 6** (Hollmén and Tresp 2000).

Sometimes, the distribution of data may be thought to change from one representation to another. This situation was examined in **Publication 6** (Hollmén and Tresp 2000), where the representation of the data switched from a continuous to a discrete case due to an artifact in the pre-processing of data. The data is augmented with its semantics and a solution to decouple the data and its semantics is presented. The semantics of the data, which is known, enables choosing the right model for the present data. The method incorporates deterministic switching between data distributions and essentially decouples the different semantics and data from each other. The temporal process of generating different data semantics becomes an integral part of the user profiles.

4.1.3 Learning by EM algorithm

Learning is the process of estimating the parameters of a model from the available set of data. In the context of probabilistic models, it is natural to consider the principle of maximum likelihood. The maximum likelihood estimate for the parameters maximizes the probability of the data for a given model. This is relatively straightforward if the variables in the model are observed, but becomes somewhat complicated, since the models of interest

here include hidden variables. This problem may be overcome by application of the EM algorithm. The EM algorithm (Dempster, Laird, and Rubin 1977; McLahlan 1996) is an iterative algorithm for estimating maximum likelihood parameters in incomplete data problems. Incomplete data means that there is a many-to-one mapping between the hidden state space and the observed measurements. Since it is impossible to recover the hidden variable, EM algorithm works with its expectation instead by making use of the measurements and the implied form of the mapping in the model. The EM algorithm is guaranteed to converge monotonically to a local maximum of the likelihood function (Dempster, Laird, and Rubin 1977; Wu 1983; Xu and Jordan 1996).

For the purpose of the EM algorithm, the expected log likelihood of the complete data (Dempster, Laird, and Rubin 1977) is introduced as

$$\begin{aligned} Q(\phi|\phi^{(old)}) &= E(logP(Y,S|\phi)|Y,\phi^{(old)}) \\ &= \int_S logP(Y,S|\phi)P(S|Y,\phi^{(old)})dS, \end{aligned} \tag{4.3}$$

where the log-likelihood of the complete data is parameterized by the free parameter value ϕ and the expectation is taken with respect to the second distribution parameterized by the current parameters $\phi^{(old)}$. In the E-step, the Q function in Equation 4.3 is computed. In Bayesian networks, this is achieved through inserting observed evidence in the network and applying propagation rules (Jensen 1996) to form the joint probability distribution of all variables or any marginalization of it. The first account that used inference techniques in the E-step appeared in (Lauritzen 1995). In the M-step, the parameter values are updated to be

$$\phi^{(new)} = \arg\max_{\phi} Q(\phi|\phi^{(old)}). \tag{4.4}$$

A solution to this maximization problem is usually found by setting the derivatives of the maximized function to zero and solving for ϕ. The application of the EM algorithm in the case of mixture models can be found in the literature (Redner and Walker 1984; Bishop 1996). Interestingly, the learning technique used in HMM (Baum 1972) turns out to be an instance of the EM algorithm. Learning in regime-switching models within the framework of maximum likelihood was formulated by Hamilton (1990, 1994). He used a regime-switching model to identify recession periods in the US economy. In **Publication 3** (Hollmén and Tresp 1999), exact inference rules for the hierarchical regime-switching model are derived from the junction tree algorithm of the Bayesian networks (Jensen 1996). A recent account on learning from data with graphical models can be found in (Heckerman 1999).

4.1.4 Finite mixture models

In finite mixture models (Everitt and Hand 1981; Redner and Walker 1984; Titterington et al. 1985), one assumes an observed variable Y that is conditioned on a discrete hidden variable S. The observed variable may be either discrete or continuous. The joint probability density is then

$$P(S, Y) = P(S)P(Y|S). \tag{4.5}$$

Integration (summation) over the hidden variable S gives an equation for calculating the likelihood of observed data in a more recognizable form as $P(Y) = \sum_{j=1}^{k} P(S = j) \prod_{i=1}^{n} p(y_i|S = j)$. Graphical illustration is shown in Figure 4.2.

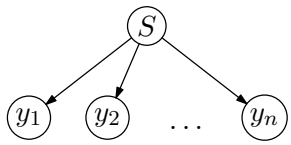

Figure 4.2: A mixture model is shown. The observed variable $Y = (y_1, \ldots, y_n)^T$ is conditioned on a discrete hidden variable S. Observed samples are assumed to be independent.

A Gaussian mixture model was used in **Publication 2** in modeling of users' recent behavior using feature data describing daily usage. After estimating a general model from a database of user data, the model is allowed to specialize to individual user profiles by estimating the mixing proportions of the component densities $P(S = j)$ on-line as more call data becomes available. Other parameters are considered to be fixed. The on-line estimation is due to Nowlan (1991). The main result of the **Publication 2** is that with this approach one is able to detect fraud accurately based on daily usage data.

4.1.5 Hidden Markov models (HMM)

A more complicated model that takes time dependencies into account is the hidden Markov model (HMM), which is widely used in sequence processing and speech recognition (Baum 1972; Juang and Rabiner 1991; Bengio 1999). Smyth et al. (1997) consider HMMs in a general framework of probabilistic independence networks and show that algorithms for inference and learning are special cases of more general class of algorithms. For a review on HMM, see (Levinson et al. 1983; Poritz 1988). These models assume a discrete, hidden state s_t, observations y_t that are conditioned on the hidden state as $P(y_t|s_t)$ and the state transitions as $P(s_t|s_{t-1})$. The joint probability

density is then

$$P(Y,S) = P(y_0,s_0)\prod_{t=1}^{T} P(s_t|s_{t-1};\theta_1)\prod_{t=1}^{T} P(y_t|s_t;\theta_2), \tag{4.6}$$

where the current state is conditionally independent of the whole history given the previous state $P(s_t|s_{t-1},s_{t-2},\ldots,s_1) = P(s_t|s_{t-1})$. This is called the Markov property, which is prevalent in many kinds of time-series models. Moreover, the current observation is conditionally independent of the whole history given the current hidden state. In essence, the state information summarizes the whole history. The graphical presentation of the HMM is shown in Figure 4.3.

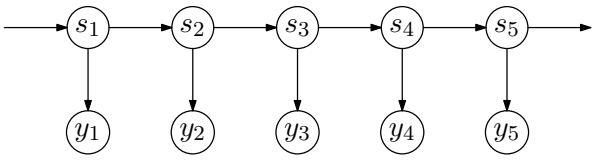

Figure 4.3: In a hidden Markov model, one assumes a hidden variable s_t that obeys transitions in time defined by $P(s_t|s_{t-1})$. The observations are conditioned on the hidden variable as $P(y_t|s_t)$.

In **Publication 6** (Hollmén and Tresp 2000), an HMM was extended to handle a switching representation between metric and event-based data. The model introduces a variable, which determines the correct interpretation of data (see Figure 4.4). The idea is to decouple the occurrence of different

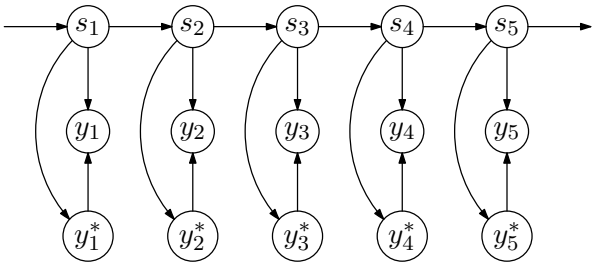

Figure 4.4: Extended version of the HMM that enables modeling of data streams switching between metric and event-based data. The introduced variable y_t^* determines the correct density to be used in interpreting the likelihood of observed data.

data semantics and the data itself. The observed data semantics expresses whether the data is to be interpreted as metric or event-based. The data semantics forms a dimension of its own in the user profile. The ideas are

illustrated in the case of hidden Markov model, for which inference and learning rules are developed.

4.1.6 Hierarchical regime-switching model

In **Publication 3** (Hollmén and Tresp 1999), a more complicated structure is used, which differs from HMM in two aspects. First, the hidden variable that develops in time has a hierarchical structure and second, the probability density for the observations is dependent on past observations. The hierarchical organization involves two layers of states, each of which develops in time according to a Markov chain and the middle layer is conditioned on the layer above. In all, the joint probability for observations and the hidden states (V in the top layer and S in the middle layer, see Figure 4.5) is

$$P(Y,S,V) = P(y_0,s_0,v_0)\prod_{t=1}^{T} P(v_t|v_{t-1};\theta_1) \times \prod_{t=1}^{T} P(s_t|v_t,s_{t-1};\theta_2)\prod_{t=1}^{T} P(y_t|s_t,y_{t-1};\theta_3). \quad (4.7)$$

The idea in regime-switching models is to model a problem domain with multiple models allowing the generating mechanism to switch from one mode of operation to another in an indeterministic fashion (Quandt 1958; Quandt and Ramsey 1972; Shumway and Stoffer 1991; Hamilton 1990; Hamilton 1994).

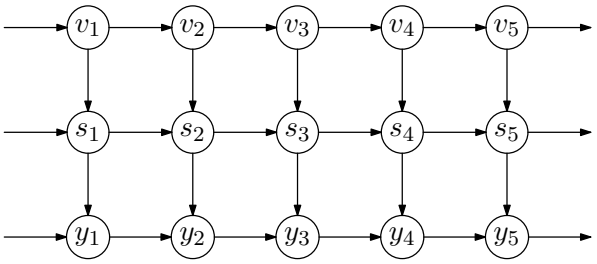

Figure 4.5: Hierarchical regime-switching model is shown. Hidden variables v and s have a hierarchical structure and middle layer is conditioned on the top layer. Furthermore, observations y_t are conditioned on a previous observation y_{t-1} and the current fraud state s_t.

In **Publication 3**, the motivation for introducing hierarchy in the model is to model fraud at different time scales. The middle layer would model fraud as expressed in the call data and the top layer would model whether the account is victimized, that is, whether the fraudster could call if he chose to. The maximum likelihood framework is used in training; additionally gradient

based training is used to enhance the discriminative nature of the model. The results demonstrate the feasibility of the methods in fraud detection. Despite the difficult set up of the problem, the model is able to detect over 90 % of the fraudsters with a false alarm probability of 2 %.

4.2 Self-Organizing Map (SOM)

4.2.1 SOM algorithm

The Self-Organizing Map (SOM) is a neural network model for the analysis and visualization of high-dimensional data. It was invented by Academician Teuvo Kohonen (1981, 1990, 1995) and is the most popular network model based on unsupervised, competitive learning. Self-Organizing Map has been used in a wide range of applications (Kaski et al. 1998). It has also been applied for the analysis of industrial processes (Kohonen, Oja, Simula, Visa, and Kangas 1996; Alhoniemi, Hollmén, Simula, and Vesanto 1999; Simula, Vesanto, Alhoniemi, and Hollmén 1999). Earlier work on process monitoring can be seen as the groundwork leading to the problem of fraud detection, which may be seen as a *user* monitoring problem. In **Publication 1**, work on the process modeling and monitoring problem is reported.

The Self-Organizing Map is a collection of prototype vectors, between which a neighborhood relation is defined. This neighborhood relation defines a structured lattice, usually a two-dimensional, rectangular or hexagonal lattice of map units. After initializing the prototype vectors with, for example, random values, training takes place. Training a Self-Organizing Map from data is divided into two steps, which are applied alternately. First, a best-matching unit (BMU) or a winner unit m^c is searched, which minimizes the Euclidean distance between a data sample x and the map units m^k

$$c = \arg\min_k \|x - m^k\|. \tag{4.8}$$

Then, the map units are updated in the *topological* neighborhood of the winner unit. The topological neighborhood is defined in terms of the lattice structure, not according to the distances between data samples and map units. The update step can be performed by applying

$$m^k(t+1) := m^k(t) + \alpha(t)h^c(t,k)[x(t) - m^k(t)], \tag{4.9}$$

where the last term in the square brackets is proportional to the gradient of the squared Euclidean distance $d(x, m^k) = \|x - m^k\|^2$. The learning rate $\alpha(t) \in [0, 1]$ must be a decreasing function of time and the neighborhood function $h^c(t, k)$ is non-increasing function around the winner unit defined in the topological lattice of map units. A good candidate is a Gaussian around the winner unit defined in terms of the coordinates r in the lattice

of neurons

$$h^c(t,k) = \exp\left(-\frac{\|r^k - r^c\|^2}{2\sigma(t)^2}\right). \quad (4.10)$$

During learning, the learning rate and the width of the neighborhood function are decreased, typically in a linear fashion. In practice, the map then tends to converge to a stationary distribution, which reflects the properties of the probability density of data.

The Self-Organizing Map may be visualized by using a unified distance matrix representation (Ultsch and Siemon 1990), where the clustering of the SOM is visualized by calculating distances between the map units locally and representing these visually with gray levels. Another choice for visualization is the nonlinear Sammon's mapping (Sammon Jr. 1969), which projects the high-dimensional map units on a plane by minimizing the global distortion of inter point distances.

4.2.2 SOM in process monitoring

Self-Organizing Map has found many applications in industrial environments (Kohonen, Oja, Simula, Visa, and Kangas 1996). Early work on monitoring the state of an industrial process is reported in (Tryba and Goser 1991; Kasslin, Kangas, and Simula 1992). The goal in process monitoring is to develop a representation of the state of an industrial process from process data and to use this representation in monitoring the current state of the process. Although conceptually operating on a different level, the problem of process monitoring is similar to the problem of monitoring users. In process monitoring, the focus is on the system level, whereas in user monitoring problems, the users are thought to form individual processes to be monitored. The problem of activity monitoring focusing on the level of users is also articulated by Fawcett and Provost (1999). The work in the area of process monitoring is reported in **Publication 1** (Alhoniemi, Hollmén, Simula, and Vesanto 1999) and also in (Simula, Alhoniemi, Hollmén, and Vesanto 1997; Simula, Vesanto, Alhoniemi, and Hollmén 1999; Simula, Ahola, Alhoniemi, Himberg, and Vesanto 1999). This work lays the groundwork for the later work in user profiling and classification problems.

In the simple example presented in **Publication 1** (Alhoniemi, Hollmén, Simula, and Vesanto 1999), a computer system is monitored in terms of its internal state measured by its central processor unit (CPU) activity as well as its network connections. The analysis begins by collecting the time-dependent measurements in a measurement vector and by pre-processing them appropriately. By training a Self-Organizing Map from the measurements, different states of the system may be visualized and the current state may be mapped to the characterized states for understanding the behavior of the process. Although the authors were ignorant of the work in intrusion

detection at the time the work was done, the work bears some analogs, at least conceptually.

4.2.3 SOM for clustering probabilistic models

Publication 4 presents a Self-Organizing Map algorithm, which enables using probabilistic models as the cluster models. In this approach the map unit indexed by k stores the empirically estimated parameter vector θ^k with an associated probabilistic model $q(x\,;\theta^k)$. For implementing a Self-Organizing Map algorithm, one needs to define a distance between the map units (i.e. the θ^k) and data. The distance between θ and a data point itself can not be defined in Euclidean space since they may have different dimensionality. The most common distance measure between probability distributions is the Kullback-Leibler distance (Bishop 1996; Ripley 1996), which relates two probability distributions. If one considers the data sample x_i to be distributed according to an unknown probability distribution $x_i \sim p(x)$ then one may approximate $p(x) \approx \delta(x - x_i)$ by placing a unit impulse $\delta(x)$ at the data point. If this expression is substituted into the Kullback-Leibler distance, one gets

$$KL(p \parallel q) = -\int p(x) \log \frac{q(x\,;\theta^k)}{p(x)} dx \Rightarrow -\log q(x_i\,;\theta^k), \tag{4.11}$$

which is the negative log probability of data for the empirical model. Thus, minimizing the Kullback-Leibler distance between the unknown true distribution that generated the data point at hand and the empirical model leads to minimizing the negative logarithm of the probability of the data with the empirical model. This justifies the use of this probability measure as a distance measure between models and data. In light of this derivation, one can derive a Self-Organizing Map algorithm for parametric probabilistic models. A winner unit indexed by c is defined by minimizing the negative log-likelihood of the empirical models for a given data point or equivalently, by searching for the maximum likelihood unit as in

$$c = \arg\min_k [-\log q(x_i\,;\theta^k)] = \arg\max_k q(x_i\,;\theta^k). \tag{4.12}$$

The update rules are based on the gradients of this likelihood in the topological neighborhood of the winner unit c as

$$\theta^k(t+1) := \theta^k(t) + \alpha(t) h^c(t,k) \frac{\partial \log q(x(t);\theta^k)}{\partial \theta^k}. \tag{4.13}$$

To illustrate the idea, an algorithm for a specific case of user profiling in mobile phone networks is derived in **Publication 4**.

4.3 Learning Vector Quantization (LVQ)

4.3.1 LVQ algorithm

The Learning Vector Quantization algorithm (Kohonen 1990; Kohonen 1995; Kohonen, Hynninen, Kangas, Laaksonen, and Torkkola 1996) estimates a classifier from labeled data samples. The classifier consists of a labeled set of codebook vectors, and the classification is based on the nearest-neighbor rule. Thus, the method does not, in contrast to traditional vector quantization or SOM, approximate the class densities, but defines the class borders by the placement of class-specific codebook vectors (Kohonen 1995). This approach is also motivated by the observation that in a discrimination task a good estimate for the class density is only needed near the class border. In the training phase, for a random sample x, there is a winner unit among the codebook vectors m^k defined by

$$c = \arg\min_k \|x - m^k\|. \tag{4.14}$$

This winner unit m^c is adapted in order to decrease the expected misclassification probability for the training set according to

$$m^c(t+1) := m^c(t) \pm \alpha(t)[x(t) - m^c(t)]. \tag{4.15}$$

The sign is chosen according to the correctness of the classification. If the label of the training sample matches that of the nearest codebook vector, the sign + is chosen, otherwise − is chosen. In LVQ, only the winner unit is updated, in contrast to the SOM algorithm, where a neighborhood function around the winner determines the map units to be updated. The class border defined by the codebook vectors and the nearest-neighbor classification rule approximates the Bayes' decision surface (Kohonen 1995).

4.3.2 LVQ for probabilistic models

In LVQ, the class border is defined by codebook vectors, which are prototypes in the input space. If representing data by prototypes is infeasible, one may replace the concept of a prototype by data generating models. This has been considered in **Publication 7** (Hollmén, Tresp, and Simula 2000), which extends the ideas in **Publication 4** (Hollmén, Tresp, and Simula 1999) to the classification domain. As far as the distance measures are concerned, the same reasoning applies to the LVQ algorithm as does for the SOM algorithm (Equation 4.11). Relating empirical models and the unknown densities behind the data samples with the Kullback-Leibler distance measure leads to the negative logarithm of the probability of data samples with the empirical models. Therefore, as in the case of the SOM algorithm, the winner search looks for a maximum likelihood unit indexed by c as in the Equation 4.12. The update is based on the gradient update of the winner unit. The

direction of the gradient update is dependent on the correctness of the classification of the sample point. If the data sample is classified correctly, the winner unit is adapted towards the sample, if incorrectly classified, the data repulses the winner unit away from the data sample:

$$\theta^c(t+1) := \theta^c(t) \pm \alpha(t)\frac{\partial \log q(x(t);\theta^c)}{\partial \theta^c}. \tag{4.16}$$

Using this approach, one may train classifiers that take advantage of the probabilistic model formulated for a user profiling problem, but which are specifically tuned for the problem of discrimination. Moreover, the probabilistic models in the codebook could be used as component densities in a mixture model, the mixing coefficients set to equal values. Such a model would produce continuous outputs on the class membership. This enables using different thresholds in tuning the classifier to best performance.

4.4 Cost-sensitive classification

In Bayes decision theory, the decision problem is posed in probabilistic terms, and it is assumed that all of the relevant probability values are known (Duda and Hart 1973). The decision goal of the application determines the decision function. For instance, minimization of the probability of misclassification as the decision goal leads to choosing the class with maximum posterior probability (Duda and Hart 1973; Schalkoff 1992). This decision function optimizes under the assumption of equal costs associated with errors, which is equivalent to optimizing the number of correct decisions. In many applications, such as in fraud detection, costs are important and should be considered. The decision goal is to minimize the costs of misclassification, which in turn leads to choosing the class with minimum conditional risk. Conditional risk for making a decision α_i may now be defined as

$$R(\alpha_i|x) = \sum_{j=1}^{n} \lambda_{ij} P(\omega_j|x). \tag{4.17}$$

which is a weighted sum of misclassification costs. This is the average cost under the uncertainty of the correct class. The misclassification costs λ_{ij} are estimates of the costs of choosing class i when class j is the true class and usually estimated as averages for an application. Pazzani et al. (1994) consider cost-sensitive classification in this framework by modifying different learning algorithms.

4.4.1 Input-dependent misclassification cost

In areas like fraud, the cost of misclassifications can not reasonably be approximated by a constant λ_{ij}, but varies from case to case. The losses are

certainly affected by service charges and the labor costs due to human processing of alarms. Furthermore, the satisfaction of a legitimate subscriber may be decreased due to the fraud accusations in the case of a false alarm. These are the motivations for a cost model, in which the misclassification costs are dependent on the input data. This has been considered in **Publication 5** (Hollmén, Skubacz, and Taniguchi 2000). The conditional risk may be formulated now as

$$R(\alpha_i|x) = \sum_{j=1}^{n} \lambda_{ij}(x) P(\omega_j|x) \tag{4.18}$$

where the $\lambda_{ij}(x)$ is the misclassification cost function taking into account the properties of the data point x (call data). The cost model may have factors such as service charges for calls or transaction costs. The corresponding decision functions may be written as a function of the posterior probability of the interesting class $P(\omega_2|x)$, for example fraud. This is shown in equation 4.19.

$$[\lambda_{12}(x) + \lambda_{21}(x)]P(\omega_2|x) - \lambda_{21}(x) \underset{\alpha_1}{\overset{\alpha_2}{\gtrless}} 0 \tag{4.19}$$

The main result of the paper is that the method performs favorably when applied to the practical problems, despite a simplified model and inaccurate class priors, making it an appealing choice for practical data mining problems.

4.5 Assessment of models

Before the developed methods are put into practice, it is important to measure their performance. In the next sections, the Receiver Operating Characteristics (ROC) curves for assessing diagnostic accuracy and cost assessment are presented.

4.5.1 Assessment of diagnostic accuracy

In the fundamental detection problem (Green and Swets 1966; Egan 1975), the task of the observer is to decide on the basis of uncertain evidence whether the stimulus consisted of a signal embedded in noise or noise alone. Observations are either *accepted* as signals in noise or *rejected* as noise alone according to a decision rule. Rephrasing this terminology from the field of psychophysics, one has a detection system (or a classifier) that on the basis of measurements, in this case call data, decides whether the calling behavior is normal or fraudulent. In the fraud detection domain, one is interested in how accurately these statements can be made.

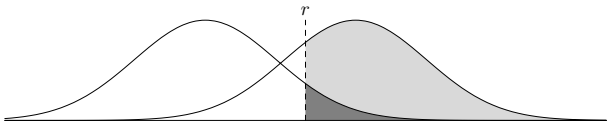

Figure 4.6: The class densities for the decision variable are shown. The dashed vertical line at r is the cut-off point for a decision. Probability of detection is marked with light gray and the probability of false alarm with dark gray. ROC curve visualizes the effect of r on these probabilities.

The evaluation must be made for each class separately, since by classifying all the cases trivially as normal a small (misleading) error rate would be achieved. This is based on the observation that fraud is indeed rare and normal behavior is dominating. Also, incorrect classifications may have different consequences. In such domains, it is natural to consider class specific assessment of the detection capability, which leads to ROC analysis (Green and Swets 1966; Egan 1975; Metz 1978; Swets 1988). ROC curve is a function that summarizes the possible performances of a detector

$$ROC = \left\{ (u,v) \middle| u = \int_r^\infty p(x|\omega_1)dx \; ; v = \int_r^\infty p(x|\omega_2)dx \right\}. \qquad (4.20)$$

It does so by varying the cut-off point of decision (threshold) along the chosen decision variable. It can be presented as a graphical plot, where the probability of detection is plotted as a function of the probability of false alarm. Formally, a ROC curve is a curve of points (u, v), where $p(x|\omega_1)$ and $p(x|\omega_2)$ are the probability densities for the decision variable. This is shown in the Equation 4.20. In this thesis, the decision variable is based on the likelihood ratio or the posterior class probabilities. ROC visualizes the trade-off between false alarms and detection, thus facilitating the choice of a decision function. Illustration of a ROC curve corresponding to the Figure 4.6 is found in Figure 4.7.

Hanley and McNeil (1982) show that the area under the ROC curve corresponds to the probability that a randomly chosen pair $(x_1 \in \omega_1, x_2 \in \omega_2)$ is correctly ordered (ranked). Hilgers (1991) presents a method to estimate the distribution-free confidence bounds of ROC curves for finite samples.

Due to the requirements on a fraud detection system, it should be assessed in terms of ROC curves. This has also been motivated by (Provost, Fawcett, and Kohavi 1998; Stolfo, Fan, Lee, and Prodromidis 1997). In practice, a fraud detection system should not produce too many alarms, because their processing is expensive. In addition, the ratio between correct alarms and all alarms should be relatively high in order to avoid inflation of alarm values.

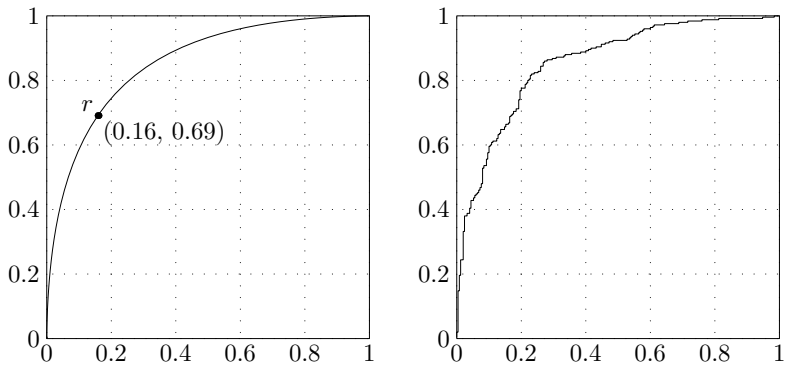

Figure 4.7: In the left panel, ROC curve for the distributions in Figure 4.6 is shown. The cut-off point r is marked in the figure, corresponding to a false alarm probability of 0.16 and a detection probability of 0.69. In the right panel, empirically estimated ROC curve for the same distributions is shown. Samples from each class (n = 250) were generated from the Gaussian distributions ($\mu_1 = 0, \mu_2 = 1.5, \sigma_1 = \sigma_2 = 1$).

4.5.2 Cost assessment

In the previous section, the accuracy of a detection system was assessed with ROC curves. Cost issues are not considered in the presentation of a ROC curve, but presenting the curve recognizes the importance of class-specific evaluation. The final goal in fraud detection is to minimize costs incurred through fraud.

Ezawa and Norton (1996) state that the cost issues are handled surprisingly little in the literature. Fawcett and Provost (1997) present cost models in fraud domain against which they assess their fraud detection system. Based on their fraud estimate, they state a fixed cost for every minute of fraudulent activity. They give cost estimates with different decision schemes with their rule-based system and compare them with trivial decision schemes such as "classify all as fraudulent" and "classify all as normal". In (Provost and Fawcett 1997), the authors present a ROC analysis in the case of non-uniform class and cost distributions.

4.5.3 Relationship between ROC analysis and cost

It is interesting to relate the cost-sensitive methods described above with ROC analysis (Green and Swets 1966; Egan 1975). The following assumes no cost for a correct classification. Whereas the ROC curve has the information about the false negatives and false positives for a given population and for varying decision functions, the standard cost-sensitive classification based on fixed misclassification costs maps these two quantities for individual data

samples to the cost space by a linear mapping. The extended method is presented in **Publication 5** (Hollmén, Skubacz, and Taniguchi 2000). It incorporates the input data in formulation of the cost model, has the same kind of mapping. This mapping is additionally parameterized by the data, enabling the costs to vary from one case to another.

4.6 Discussion

This chapter introduced background to put the methods of the thesis in the right context and to appreciate the novelties in them.

The first developed methods based on adaptive Gaussian mixture models presented in **Publication 2** (Taniguchi, Haft, Hollmén, and Tresp 1998) approached the problem with an adaptive, user-specific modeling. This may be prohibitive in the presence of many users. Whereas the decision-making should be retained at the user level, modeling effort may be done either at the user level, user profile level or class level. The methods at the user level could be made computationally lighter by applying the methods in **Publication 4** (Hollmén, Tresp, and Simula 1999) in defining a small number of Gaussian mixture models, which in turn would model prototypical user profiles. These models could be further enhanced for discrimination by applying discriminative learning of **Publication 7**. The discriminative training procedure in **Publication 7** could also stand as a method of its own, provided there are enough labeled training samples. In retrospect, modeling based on several prototypes should be preferred to fully adaptive modeling approach in the problem of fraud detection.

The applicability of the methods presented in **Publication 3** (Hollmén and Tresp 1999) are limited by the computational resources, but the work demonstrates that the dynamic modeling of fraud is a successful approach, despite the demanding problem formulation and the sceptical view of some authors. Since the problem representation is rather extreme as the resolution used in describing the time-series of calling data is one minute, the method is computationally quite demanding.

The ideas concerning the issue of switching representations of observed data considered in **Publication 6** (Hollmén and Tresp 2000) could be introduced in any generative model, in finite mixture models, for example.

The methods in the **Publication 5** (Hollmén and Tresp 2000) formulated a cost model that was particularly suitable for the problem of fraud detection. This approach is compatible with all methods producing probabilistic outputs that fit the framework of Bayes detection theory.

Chapter 5

Conclusions

5.1 Summary

User profiling and classification are important tasks in data intensive environments where the behavior of a heterogeneous mass of users is to be understood or where computer assisted decision making is sought for. Fraud detection is a prime example of this kind of problem.

A fraud detection system attempts to discover illegitimate behavior. The system cannot directly observe the intentions of the mobile phone subscribers, but works rather on their actions, that is, their calling behavior. The calling behavior is collectively described by the subscriber's call data and is used in this thesis as a basis for modeling. The use of contextual behavior is further motivated by Davis and Goyal (1993), who state that there is nothing about any one call itself that proves incontrovertibly that it is fraudulent. The goal for the learning methods in this thesis is to learn user profiles from the call data in order to make plausible decisions about fraud occurring.

The methods presented in this thesis learn to detect fraud from partially labeled data, that is, it is known that an account is defrauded but not exactly when. The data is thus a mixture of normal and fraudulent data with an unknown mixing mechanism. No other work known to the author solves the problem of learning fraud models from data that is partially labeled in the mixture setting. This approach provides an economic aspect to learning to detect fraud. Representation of the data is also an important issue. In any data representation, however, there is a compromise between the average latency time for detection and richness of description. This thesis has used representations ranging from instantaneous calling behavior to features calculated over one day as representations of data. Also, modeling changing data semantics was presented.

The models used in fraud detection were probabilistic models and neural networks. The ability to learn from data was considered an important asset

of these models, as was the capability to process uncertainty, which is present in the fraud domain. Some variations concerning how many models per class should be allocated were considered. Modeling was performed on user level, user profile level and class level, of which the user profile level was seen to be the most appropriate. Discriminative training was also considered for tuning the models for best diagnostic accuracy.

A cost model for making cost-sensitive decisions was also discussed. As are the decisions made, so should also the cost considerations be made on an individual basis. Minimizing conditional risk of decisions under uncertainty of the correct class was taken as the criterion for the classification. Asserting fixed misclassification costs between classes only takes into account the average risk for the given domain, which was the main motivation for extending the cost model to include the effect of input data in the misclassification costs. It was shown that these cost models may naturally be defined in the domain of fraud detection.

The results in this thesis in terms of detection performance are comparable to or better than other works published in the field. As a rough measure of state-of-the-art performance, the detection system should detect most of the fraudsters, but more importantly, false alarm probabilities should be below 2 or 3 percent. Otherwise, as the population of mobile phone users may be large, the absolute number of alarms is beyond control. Ultimately, the requirements are set by the size of the subscriber population, prevalence of fraud, and the upper limit on alarms determined by organizational resources for processing them. As noted in the review part, comprehensive comparisons are difficult to make since there is no common basis for evaluation and the environment for which they were developed may differ substantially.

The methods are shown to be effective in detecting fraudulent behavior by empirically testing the methods with data from real mobile communications networks. The presented solutions to fraud detection lay the groundwork for detection methodologies to be used in an operational fraud detection system.

5.2 Further work

Further research within fraud domain concentrates on combining outputs of different expert models as presented in (Jacobs, Jordan, Nowlan, and Hinton 1991; Jacobs 1995). Also, kernel-based approaches, such as those presented by Jaakkola and Haussler (1999) could be interesting. While they are using a non-parametric base for defining the kernels, the methods in **Publication 7** could be used to learn a fixed kernel base with only a few kernels. Further work should also examine the transferability of the models to different networks. Also, inclusion of data (observed) describing the social status of the mobile phone subscriber should be considered.

The presented methods for fraud detection can be seen as solutions to a specific user profiling problem in fraud detection. Several similar problems exist, for instance in identification of customer groups in marketing, modeling dynamic navigation patterns in hypertext documents, product design for a heterogeneous mass of customers. All of these problems call for modeling user profiles from an available set of data for a specific purpose. In the age of computerization, collection and storage of data have become commonplace. Large databases contain billions of records of data that are not informative as such, but are useful after considerable data analysis process. In management science, total quality management and associated methodologies have stressed the importance of fact based decision making, and orderly collection and analysis of data. Ability to learn representations from large databases has profound consequences and will have an impact on how decisions are made.

Chapter 6

Publications

6.1 Contents of the publications

Publication 1 (Alhoniemi, Hollmén, Simula, and Vesanto 1999) presents work in the area of process monitoring and modeling with the Self-Organizing Map. Training techniques of the Self-Organizing Map are reviewed and industrial applications in monitoring of a pulp process, modeling of steel production, and in analysis of paper industry are reported. The application used as an example involves monitoring a computer system in terms of its internal state and its network connections.

Publication 2 (Taniguchi, Haft, Hollmén, and Tresp 1998) presents three methods for fraud detection. Firstly, a feed-forward neural network is used in classification of users to normal and fraudulent classes based on summary statistics over a time period. Secondly, user behavior is modeled with an adaptive Gaussian mixture model, which is used in a novelty detection fashion to detect sudden changes from the past behavior. This constitutes the contribution of the present author. Thirdly, two Bayesian networks are formulated by an expert to reflect domain knowledge about fraudulent and normal behavior. The outputs from these networks are combined with the Bayes's rule. The two latter methods are based on features calculated over a period of one day. For the methods presented in this paper, a patent (Taniguchi, Haft, Hollmén, and Tresp 1997) has been granted.

Publication 3 (Hollmén and Tresp 1999) uses a hierarchical regime-switching model in detection of fraud. Learning is based on the EM algorithm; inference rules are derived from the junction tree algorithm (Jensen 1996). In addition to unsupervised learning, the models are fine-tuned using supervised learning to improve the discriminative performance of the model. The calling data is represented as a binary time-series, which has a high sampling rate. This work is a step towards real-time detection of fraud. The learning procedure does not require fully labeled accounts, but works with partially labeled data as described in Chapter 3.

Publication 4 (Hollmén, Tresp, and Simula 1999) develops methods to cluster probabilistic models with the Self-Organizing Map algorithm. The standard Self-Organizing Map algorithm is not suitable for the task, since it uses Euclidean distance as an error measure, which can not sensibly be defined between time-series and probabilistic models. On the contrary, parameters of probabilistic models are stored in map units and a likelihood based distance measure is defined between data and map units. Update equations are derived from the gradients of likelihood; additional parameterization is introduced to handle the constraints on the parameters. A softmax layer is used to map the unconstrained parameters to the constrained parameter space. In experiments, the approach is used to model calling behavior in mobile communications networks with dynamic models.

Publication 5 (Hollmén, Skubacz, and Taniguchi 2000) presents cost models for fraud detection. It extends the standard case of considering the types of misclassification to have impact on the costs to include the data itself to have influence on the costs. This is important in fraud detection, where the call data directly influences the losses, and where the type of misclassification only expresses the average costs involved. The cost model has components for the connection-based tariffs and a transaction cost for examining accounts further. Experiments compare the performance of the new approach with standard approaches under varying assumptions.

Publication 6 (Hollmén and Tresp 2000) presents an extension of a hidden Markov model to incorporate changing data representations, where data switches between event-based and continuous representations in time. Data and its representation are decoupled from each other by introducing an additional variable for the semantics of the data that determines the appropriate type of model to be used in interpreting the data. Furthermore, the occurrence of different semantics is dependent on the hidden variable. Inference and learning rules are developed for this extension and experiments in a user profiling problem are reported.

Publication 7 (Hollmén, Tresp, and Simula 2000) derives the Learning Vector Quantization (LVQ) algorithm for probabilistic models. In the standard LVQ algorithm, the class specific codebook vectors are expressed as prototypes in the input space and in the lookup phase, a nearest neighbor rule is used in classification. In the presence of complex data, such as time series, this may be not be feasible. The use of probabilistic models together with LVQ enables learning generative models for discrimination. These data generating models define a class border together with a maximum likelihood classification rule. The implementation of the algorithm when the models involve hidden variables is further discussed. The experiments illustrate the classifier in a user classification problem in fraud detection.

6.2 Contributions of the author

In **Publication 1**, the author was responsible for the work reported in the section on process modeling using the Self-Organizing Map and the case study on steel production. The example on monitoring a computer in terms of its internal state and its network connection was conceptualized by the author and the data for the example was acquired by the author. In **Publication 2**, the author was responsible for the section on novelty detection with Gaussian mixtures. The ideas were invented by the author and experiments were made by the author. Also, writing the paper was coordinated by the author. In **Publication 3**, the author was responsible for the representation of the problem and the experiments. The inference and learning rules were developed jointly with the second author, with whom the paper was also jointly written. In **Publication 4**, the author was responsible for the ideas and the experiments. The paper was written by the author and edited by the co-authors. In **Publication 5**, the ideas, experiments and writing the paper were a joint effort of the first and the second author. In **Publication 6** and **Publication 7**, the author was responsible for the ideas and performed the experiments. The author was responsible for writing the papers, the co-authors of the papers were responsible for editing. The invention in **Publication 5** has been filed as a patent (Taniguchi, Haft, Hollmén, and Tresp 1998), others have been filed as inventions and are considered for filing as a patent application.

6.3 Errata

Equation (3) in the **Publication 2** (Taniguchi, Haft, Hollmén, and Tresp 1998) does not imply standardized values of $P(j)$ so that $\sum_j P(j) = 1$. The equation should read $P(j)^{new} = \alpha P(j)^{old} + (1 - \alpha)P(j|x)$.

References

Aleskerov, E., B. Freisleben, and B. Rao (1997). CARDWATCH: A neural network based database mining system for credit card fraud detection. In *Proceedings of the IEEE/IAFE 1997 Conference on Computational Intelligence for Financial Engineering (CIFEr)*, pp. 220–226. IEEE Press.

Alhoniemi, E., J. Hollmén, O. Simula, and J. Vesanto (1999). Process monitoring and modeling using the self-organizing map. *Integrated Computer Aided Engineering 6*(1), 3–14.

Allen, P., R. McKendrick, C. Scott, M. Buonanno, P. Mostacci, C. Naldini, V. Scuderi, and P. Stofella (1996). Interactive anomaly detection in large transaction history databases. In *High-Performance Computing and Networking. International Conference and Exhibition HPCN 1996 Proceedings*, pp. 143–149.

Anderson, J. P. (1980). Computer security threat monitoring and surveillance. Technical report, James P. Anderson Co.

Barney, L. (1995). Detecting trading fraud. *Wall Street & Technology 12*(11), 40.

Barson, P., S. Field, N. Davey, G. McAskie, and R. Frank (1996). The detection of fraud in mobile phone networks. *Neural Network World 6*(4), 477–484.

Baum, L. E. (1972). An inequality and associated maximation technique in statistical estimation for probabilistic functions of markov processes. *Inequalities 3*, 1–8.

Bengio, Y. (1999). Markovian models for sequential data. *Neural Computing Surveys 2*, 129–162.

Bishop, C. (1996). *Neural Networks in Pattern Recognition*. Oxford Press.

Burge, P. and J. Shawe-Taylor (1996). Frameworks for fraud detection in mobile telecommunications networks. In *Proceedings of the Fourth Annual Mobile and Personal Communications Seminar, University of Limerick*.

Burge, P. and J. Shawe-Taylor (1997). Detecting cellular fraud using adaptive prototypes. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 9–13. AAAI Press.

Burge, P., J. Shawe-Taylor, Y. Moreau, H. Verrelst, C. Störmann, and P. Gosset (1997). BRUTUS - a hybrid detection tool. In *Proceedings of ACTS Mobile Telecommunications Summit, Aalborg, Denmark*.

Chan, P. K. and S. J. Stolfo (1998). Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In R. Agrawal, P. Stolorz, and G. Piatetsky-Shapiro (Eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, pp. 164–168.

Cherkassky, V. and F. Mulier (1998). *Learning from data: Concepts, Theory and Methods*. John Wiley & Sons.

Connor, J. T., L. R. Brothers, and J. Alspector (1995). Neural network detection of fraudulent calling card patterns. In *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications, 2*, pp. 363–370. Laurence Erlbaum Associates.

Cowell, R., A. Dawid, S. Lauritzen, and D. Spiegelhalter (1999). *Probabilistic Networks and Expert Systems*. Statistics for Engineering and Information Science. Springer-Verlag.

Cox, K. C., S. G. Eick, G. J. Wills, and R. J. Brachman (1997). Visual data mining: recognizing telephone calling fraud. *Data mining and Knowledge Discovery 1*(2), 225–231.

Curet, O., M. Jackson, and A. Tarar (1996). Designing and evaluating a case-based learning and reasoning agent in unstructured decision making. In *1996 IEEE International Conference on Systems, Man and Cybernetics. Information Intelligence and Systems.*, Volume 4, pp. 2487–2492.

Davis, A. B. and S. K. Goyal (1993). Management of cellular fraud: Knowledge-based detection, classification and prevention. In *Proceedings of the 13th International Conference on Artificial Intelligence, Expert Systems and Natural Language, Avignon, France*, Volume 2, pp. 155–164.

Dempster, A. P., N. Laird, and D. Rubin (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B 39*, 1–38.

Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering SE-13*, 222–232.

Dorronsoro, J. R., F. Ginel, C. Sánchez, and C. S. Cruz (1997). Neural fraud detection in credit card operations. *IEEE Transactions on Neural Networks 8*(4), 827–834.

Duda, R. O. and P. E. Hart (1973). *Pattern Recognition and Scene Analysis*. John Wiley & Sons.

DuMouchel, W. and M. Schonlau (1998). A fast intrusion detection algorithm based on hypothesis testing of command transition probabilities. In R. Agrawal and P. Stolorz (Eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pp. 189–193. AAAI Press.

Egan, J. (1975). *Signal Detection Theory and ROC Analysis*. New York: Academic Press.

European Telecommunications Standards Institute (1998). Digital cellular telecommunications system (Phase 2); Event and call data (GSM 12.05 version 4.3.1). European Telecommunication Standard ETS 300 616.

Everitt, B. and D. Hand (1981). *Finite Mixture Distributions*. Monographs on Applied Probability and Statistics. Chapman and Hall.

Ezawa, K., M. Singh, and S. Norton (1996). Learning goal oriented bayesian networks for telecommunications risk management. In *Proceedings of the 13th International Conference on Machine Learning*, pp. 139–147. Morgan Kaufmann.

Ezawa, K. J. (1995). Fraud/uncollectible debt detection using a bayesian network based learning system: A rare binary outcome with mixed data structures. In *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence*, pp. 157–166. Morgan Kaufmann.

Ezawa, K. J. and S. W. Norton (1996). Constructing bayesian networks to predict uncollectible telecommunications accounts. *IEEE Expert 11*(5), 45–51.

Fanning, K., K. O. Cogger, and R. Srivastava (1995). Detection of management fraud: a neural network approach. *International Journal of Intelligent Systems in Accounting, Finance and Management 4*(2), 113–126.

Fawcett, T. and F. Provost (1996). Combining data mining and machine learning for effective user profiling. In E. Simoudis, J. Han, and U. Fayyad (Eds.), *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96)*, pp. 8–13. AAAI Press.

Fawcett, T. and F. Provost (1997). Adaptive fraud detection. *Journal of Data Mining and Knowledge Discovery 1*(3), 291–316.

Fawcett, T. and F. Provost (1999). Activity monitoring: Noticing interesting changes in behavior. In S. Chaudhuri and D. Madigan (Eds.), *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 53–62.

Field, S. and P. Hobson (1997). Techniques for telecommunications fraud management. In J. Alspector, R. Goodman, and T. X. Brown (Eds.), *Proc. Int. Workshop on Applications of Neural Networks to Telecommunications 3*, Hillsdale, NJ, pp. 107–115. Lawrence Erlbaum.

Fox, K. L., R. R. Henning, J. H. Reed, and R. P. Simonian (1990). A neural network approach towards intrusion detection. In *Proc. 13th National Computer Security Conference. Information Systems Security. Standards - the Key to the Future*, Volume I, Gaithersburg, MD, pp. 125–134. NIST.

Frank, J. (1994). Artififcial intelligence and intrusion detection: Current and future directions. In *National Computer Security Conference*, Volume 1, pp. 22–33.

Ghosh, S. and D. L. Reilly (1994). Credit card fraud detection with a neural network. In *Proceedings of the Twenty-Seventh Hawaii International Conference on System Sciences*, pp. 621–630. IEEE Computer Society Press.

Glasgow, B. (1997). Risk and fraud in the insurance industry. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 20–21. AAAI Press.

Green, D. and J. Swets (1966). *Signal Detection Theory and Psychophysics*. Wiley, New York.

Hamilton, J. D. (1990). Analysis of time series subject to changes in regime. *Journal of Econometrics 45*, 39–70.

Hamilton, J. D. (1994). *Time Series Analysis*. Princeton University Press.

Hanagandi, V., A. Dhar, and K. Buescher (1996). Density-based clustering and radial basis function modeling to generate credit card fraud scores. In *Proceedings of the IEEE/IAFE 1996 Conference on Computational Intelligence fo Financial Engineering (CIFEr)*, pp. 247–251. IEEE Press.

Hanley, J. A. and B. J. McNeil (1982). The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology 143*(1), 29–36.

He, H., J. Wang, W. Graco, and S. Hawkins (1997). Application of neural networks to detection of medical fraud. *Expert Systems with Applications 13*(4), 329–336.

Heckerman, D. (1999). A tutorial on learning with bayesian networks. In M. I. Jordan (Ed.), *Learning in Graphical Models*, pp. 301–354. MIT Press.

Hilgers, R. (1991). Distribution-free confidence bounds for ROC curves. *Methods of Information on Medicine 30*(2), 96–101.

Hoath, P. (1998). Telecoms fraud, the gory details. *Computer Fraud & Security 20*(1), 10–14.

Hollmén, J., M. Skubacz, and M. Taniguchi (2000). Input dependent misclassification costs for cost-sensitive classification. In N. Ebecken and C. Brebbia (Eds.), *DATA MINING II — Proceedings of the Second International Conference on Data Mining 2000*, pp. 495–503. WIT Press.

Hollmén, J. and V. Tresp (1999). Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. In M. Kearns, S. Solla, and D. Cohn (Eds.), *Advances in Neural Information Processing Systems 11: Proceedings of the 1998 Conference (NIPS'11)*, pp. 889–895. MIT Press.

Hollmén, J. and V. Tresp (2000). A hidden markov model for metric and event-based data. In *Proceedings of EUSIPCO 2000 — X European Signal Processing Conference*, Volume II, pp. 737–740.

Hollmén, J., V. Tresp, and O. Simula (1999). A self-organizing map for clustering probabilistic models. In *Proceedings of the Ninth International Conference on Artificial Neural Networks (ICANN'99)*, Volume 2, pp. 946–951. IEE.

Hollmén, J., V. Tresp, and O. Simula (2000). A learning vector quantization algorithm for probabilistic models. In *Proceedings of EUSIPCO 2000 — X European Signal Processing Conference*, Volume II, pp. 721–724.

Howard, P. and P. Gosset (1998). D20 — project final report and results of trials. ASPeCT: Advanced Security for Personal Communications Technologies, Report AC095/VOD/W31/DS/P/20/E.

Jaakkola, T. and D. Haussler (1999). Exploiting generative models in discriminative classifiers. In M. Kearns, S. Solla, and D. Cohn (Eds.), *Advances in Neural Information Processing Systems: Proceedings of the 1998 Conference (NIPS'11)*, pp. 487–493. MIT Press.

Jacobs, R. (1995). Methods for combining experts' probability assessment. *Neural Computation 7*(5), 867–888.

Jacobs, R. A., M. I. Jordan, S. J. Nowlan, and G. E. Hinton (1991). Adaptive mixtures of local experts. *Neural Computation 3*, 79–87.

Jensen, D. (1997). Prospective assessment of AI technologies for fraud detection: A case study. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 34–38. AAAI Press.

Jensen, F. V. (1996). *An Introduction to Bayesian Networks*. UCL Press.

Johnson, M. (1996). Cause and effect of telecoms fraud. *Telecommunication (International Edition) 30*(12), 80–84.

Juang, B. and L. Rabiner (1991). Hidden markov models for speech recognition. *Technometrics 33*(3), 251–272.

Kaski, S., J. Kangas, and T. Kohonen (1998). Bibliography of self-organizing map (SOM) papers: 1981-1997. *Neural Computing Surveys 1*, 102–350.

Kasslin, M., J. Kangas, and O. Simula (1992). Process state monitoring using self-organizing maps. In I. Aleksander and J. Taylor (Eds.), *Artificial Neural Networks, 2*, Volume II, Amsterdam, Netherlands, pp. 1531–1534. North-Holland.

Kohonen, T. (1981). Automatic formation of topological maps of patterns in a self-organizing system. In E. Oja and O. Simula (Eds.), *Proceedings of The Second Scandinavian Conference on Image Analysis*, pp. 214–220.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE 78*(9), 1464–1480.

Kohonen, T. (1995). *Self-Organizing Maps*. Springer-Verlag.

Kohonen, T., J. Hynninen, J. Kangas, J. Laaksonen, and K. Torkkola (1996). LVQ_PAK: The learning vector quantization package. Technical Report A30, Helsinki University of Technology, Laboratory of Computer and Information Science.

Kohonen, T., E. Oja, O. Simula, A. Visa, and J. Kangas (1996). Engineering applications of the self-organizing map. *Proceedings of the IEEE 84*(10), 1358–84.

Kumar, S. (1995). *Classification and detection of computer intrusions*. Ph. D. thesis, Purdue University.

Lane, T. and C. E. Brodley (1997). Sequence matching and learning in anomaly detection for computer security. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 43–49. AAAI Press.

Lane, T. and C. E. Brodley (1998). Approaches to online learning and concept drift for user identification in computer security. In R. Agrawal

and P. Stolorz (Eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pp. 259–263. AAAI Press.

Lauritzen, S. L. (1995). EM algorithm for graphical association models with missing data. *Computational Statistics & Data Analysis 19*, 191–201.

Leonard, K. J. (1993). Detecting credit card fraud using expert systems. *Computers and Industrial Engineering 25*(1–4), 103–106.

Levinson, S., L. Rabiner, and M. Sondhi (1983). An introduction to the application of the theory of probabilistic functions of a markov process to automatic speech recognition. *The Bell System Technical Journal 62*(4), 1035–1074.

Lunt, T. (1988). Automated audit trail analysis and intrusion detection: A survey. In *Proceedings of the 11th National Computer Security Conference*, pp. 65–73.

Lunt, T. F. (1990). IDES: An intelligent system for detecting intruders. In *Proceedings of the Symposium on Computer Security (CS'90), Rome, Italy*, pp. 110–121.

Lunt, T. F. (1993). A survey of intrusion detection techniques. *Computers & Security 12*(4), 405–418.

Major, J. A. and D. R. Riedinger (1992). EFD: A hybrid knowledge/statistical based system for the detection of fraud. *International Journal of Intelligent Systems 7*(7), 687–703.

McLahlan, G. J. (1996). *The EM Algorithm and Extensions*. Wiley & Sons.

Menkus, B. (1998). Some management-directed fraud incidents. *EDPACS 25*(10), 14–16.

Metz, C. E. (1978). Basic principles of ROC analysis. *Seminars in Nuclear Medicine VIII*(4), 283–298.

Moreau, Y., B. Preenel, P. Burge, J. Shawe-Taylor, C. Störmann, and C. Cooke (1996). Novel techniques for fraud detection in mobile telecommunication networks. In *Proceedings of ACTS Mobile Telecommunications Summit, Granada, Spain*.

Moreau, Y. and J. Vandewalle (1997). Fraud detection in mobile communications networks using supervised neural networks. In *Proceedings of SNN'97, Europe's Best Neural Networks Practice*. World Scientific.

Moreau, Y., H. Verrelst, and J. Vandewalle (1997). Detection of mobile phone fraud using supervised neural networks: A first prototype. In *International Conference on Artificial Neural Networks Proceedings (ICANN'97)*, pp. 1065–1070.

Mukherjee, B., L. T. Heberlein, and K. N. Levitt (1994). Network intrusion detection. *IEEE Network 8*(3), 26–41.

Northcutt, S. (1999). *Network Intrusion Detection — An Analyst's Handbook*. New Riders Publishing.

Nowlan, S. (1991). *Soft Competitive Adaptation: Neural Network Learning Algorithms based on Fitting Statistical Mixtures*. Ph. D. thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh.

O'Shea, D. (1997). Beating the bugs: Telecom fraud. *Telephony 232*(3), 24.

Parker, T. (1996). The twists and turns of fraud. *Telephony 231*(supplement issue), 18–21.

Pazzani, M., C. Merz, P. Murphy, K. Ali, T. Hume, and C. Brunk (1994). Reducing the misclassification costs. In *Proceedings of the 11th International Conference on Machine Learning*, pp. 217–225. Morgan Kaufmann.

Pequeno, K. A. (1997). Real-time fraud detection: Telecom's next big step. *Telecommunications (Americas Edition) 31*(5), 59–60.

Poritz, A. B. (1988). Hidden markov models: A guided tour. In *Proceedings of the IEEE International conference of Acoustics, Speech and Signal Processing (ICASSP'88)*, pp. 7–13.

Provost, F. and T. Fawcett (1997). Analysis and visualization of classifier performance with nonuniform class and cost distributions. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 57–63. AAAI Press.

Provost, F., T. Fawcett, and R. Kohavi (1998). The case against accuracy estimation for comparing induction algorithms. In *Proceedings of the Fifteenth International Conference on Machine Learning*, pp. 445–453. Morgan Kaufmann Publishers.

Quandt, R. (1958). The estimation of parameters of linear regression system obeying two separate regimes. *J. Am. Stat.Assoc. 53*, 873–880.

Quandt, R. and J. Ramsey (1972). A new approach to estimating switching regression. *Journal of American Statistical Society 67*(338), 306–310.

Redner, R. and H. Walker (1984). Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review 26*(2), 195–234.

Ripley, B. D. (1996). *Pattern Recognition and Neural Networks*. Cambridge University Press.

Rosenberg, E. and A. Gleit (1994). Quantitative methods in credit management: A survey. *Operations Reserach 42*(4), 589–613.

Ryan, J., M.-J. Ling, and R. Miikkulainen (1997). Intrusion detection with neural networks. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 72–77. AAAI Press.

Sammon Jr., J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers C-18*(5), 401–409.

Schalkoff, R. J. (1992). *Pattern Recognition: Statistical, Structural and Neural approahes*. John Wiley & Sons.

Schuerman, T. (1997). Risk management in the financial services industry: Through a statistical lens. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 78–82. AAAI Press.

Shortland, R. and R. Scarfe (1994). Data mining applications in BT. *BT Technology Journal 12*(4), 17–22.

Shumway, R. and D. Stoffer (1991). Dynamic linear models with switching. *Journal of the American Statistical Association 86*(415), 763–769.

Simula, O., J. Ahola, E. Alhoniemi, J. Himberg, and J. Vesanto (1999). Self-organizing map in analysis of large-scale industrial systems. In E. Oja and S. kaski (Eds.), *Kohonen Maps*, pp. 375–387. Elsevier.

Simula, O., E. Alhoniemi, J. Hollmén, and J. Vesanto (1997). Analysis of complex systems using the self-organizing map. In *Proceedings of the 1997 International Conference on Neural Information Processing and Intelligent Information Systems (ICONIP'97)*, Volume 2, pp. 1313–1317. Springer.

Simula, O., J. Vesanto, E. Alhoniemi, and J. Hollmén (1999). *Neuro-Fuzzy Tools and Techniques*, Chapter Analysis and Modeling of Complex Systems Using the Self-Organizing Map, pp. 3–22. Physica Verlag (Springer Verlag).

Smyth, P., D. Heckerman, and M. I. Jordan (1997). Probabilistic independence networks for hidden markov probability models. *Neural Computation 9*(2), 227–269.

Sokol, L. (1998). Using data mining to support health care fraud detection. In *PADD98. Procedings of the Second International Conference on the Practical Application of Knowledge Discovey and Data Mining*, pp. 75–82.

Stolfo, S. J., D. W. Fan, W. Lee, and A. L. Prodromidis (1997). Credit card fraud detection using meta-learning: Issues and initial results. In *Proceedings of AAAI-97 Workshop on AI Approaches to Fraud Detection & Risk Management*, pp. 83–90. AAAI Press.

Swets, J. A. (1988). Measuring the accuracy of diagnostic systems. *Science 240*, 1285–1293.

Tan, K. (1995). The application of neural networks to UNIX computer security. In *1995 IEEE International Conference on Neural Networks*, pp. 476–481. IEEE Press.

Taniguchi, M., M. Haft, J. Hollmén, and V. Tresp (1997). Erkennung eines betrügerischen anrufs mittels eines neuronalen netzes. Patent DE 197 29 630 A1.

Taniguchi, M., M. Haft, J. Hollmén, and V. Tresp (1998). Fraud detection in communications networks using neural and probabilistic methods. In *Proceedings of the 1998 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'98)*, Volume II, pp. 1241–1244.

Titterington, D., A. Smith, and U. Makov (1985). *Statistical analysis of finite mixture distributions*. Wiley Series in Probability and Mathematical Statistics. John Wiley & Sons.

Tryba, V. and K. Goser (1991). Self-organizing feature maps for process control in chemistry. In T.Kohonen, K. Mäkisara, O.Simula, and J.Kangas (Eds.), *Artificial Neural Networks*, pp. 847–852. North-Holland.

Ultsch, A. and H. Siemon (1990). Kohonen's self-organizing maps for exploratory data analysis. In *Proceedings of the International Neural network Conference (INNC'90)*, pp. 305–308. Kluwer.

Wu, C. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics 11*(1), 95–103.

Xu, L. and M. Jordan (1996). On convergence properties for EM algorithm for gaussian mixtures. *Neural Computation 8*, 129–151.

Yuhas, B. (1993). Toll-fraud detection. In *Proceedings of the International Workshop on Applications of Neural Networks to Telecommunications (IWANNT'93)*, pp. 239–244.