"Non importa la forma...
nè il vestito che indossi,
nè come, quando, dove
nè il nome con cui ti chiamano...
il mio affetto sarà sempre con te,
fino al giorno che verrà...
quando i sogni ci riuniranno ancora.
Grazie per essere sempre stato con noi."

- Floricienta Bertotti

To my father, *Francesco Saverio.*

Acknowledgements

My supervisors, Prof. Domenico Saccà and Dr. Giuseppe Manco, have been very encouraging and helpful. My deepest appreciation to them for ensuring that resources are available for me, and also consistently giving me sound, valuable advices.

I wish to thank all my collegues of Institute of High Performance Computing and Networking of the National Research Council (ICAR-CNR) for their help during the period of research at ICAR, in particular the ing. Giuseppe Papuzzo, the ing. Vincenzo Russo and the ing. Ettore Ritacco.

My parents made many personal sacrifices for me to get an overseas education. At this point in time, i hope that enough has been done in my studies to make them feel proud, not only of me, but also of themselves.

Finally, dad know that wherever you are, i miss you.

Preface

Due to the dramatic increase of fraud which results in loss of billions of dollars worldwide each year, several modern techniques in detecting fraud are continually evolved and applied to many business fields. Fraud detection involves monitoring the behavior of populations of users in order to estimate, detect, or avoid undesirable behavior.

The purpose of this dissertation is to determine the most appropriate data mining methodology, methods, techniques and tools to extract knowledge or insights from enormous amounts of data to detect fraudolent behaviour. Fraud detection in VAT context is used as the application domain.

The focus is on overcoming the technical problems and alleviating the practical problems of data mining in fraud detection. The technical obstacles are due to imperfect, highly skewed data, and hard-to-interpret predictions. The practical barriers are caused by the dearth in domain knowledge, many evolving fraud patterns, and the weaknesses in some evaluation metrics.

Several methods and techniques are introduced to solve these problems. In particular first, it was proposed a new methodology to individuate fraudsters, called *Sniper*, that aims to maximize the "quality" of those selected through the use of rule-based systems and ensemble methods. This method was generalized in order to identify "exceptional" fraudolent behaviours in application domains in which these behaviors are not labeled but are highlighted by a function on a continuous range of values. Finally was presented a hierarchical classification framework that works on a more general classification area i.e. imprecise enviroments featured by noise, low occurrence of some cases of interest and low class separability.

The preliminary results in the VAT data set confirm the effectiveness of the *Sniper* methodology to predict anomalous behaviours, while a massive experimentation shows that the use of the hierarchical framework improves signifi-

cantly the accuracy on the primary class in imprecise enviroments.

This thesis demonstrates that these techniques have the potential to significantly reduce loss from illegitimate behaviour.

Università della Calabria — *Massimo Guarascio*
November 2010 — *First name Surname*

Contents

List of Figures

List of Tables

1

Introduction

The world is overwhelmed with millions of inexpensive gigabyte disks containing terabytes of data. It is estimated that these data stored in all corporate and government databases worldwide double every twenty months. The types of data available, in addition to the size, are also growing at an alarming rate. Some relevant examples can put this situation into perspective:

- In the United States (US), DataBase Technologies (DBT) Online Incorporated contains four billion records used by its law enforcement agencies. The Insurance Services Office Incorporated (ISO) claim search database contains over nine billion US claim records, with over two billion claims records added annually [32].

- In Australia, the Insurance Reference Service (IRS) industry database consists of over thirteen million individual insurance claims records [32]. The Law Enforcement Assistance Program (LEAP) database for the Victorian police in Australia details at least fourteen million vehicle, property, address and offender records, with at least half a million criminal offences and incidents added annually.

- In the last 15 years, eBay grew from a simple website for online auctions to a full-scale e-commerce enterprise that processes petabytes of data. A problem faced by all e-commerce companies is misuse of their systems and, in some cases, fraud. For example, sellers may deliberately list a product in the wrong category to attract user attention, or the item sold is not as the seller described it.

- Private firms sell all types of data on individuals and companies, often in the forms of demographic, real estate, utility usage, telecom usage, automobile, credit, criminal, government, and Internet data (Infoglide Software

Corporation, 2002).

This results in a data rich but information poor situation where there is a widening gap between the explosive growth of data and its types, and the ability to analyse and interpret it. Hence there is a need for a new generation of automated and intelligent tools and techniques [53], to look for patterns in data. These patterns can lead to new insights, competitive advantages for business, and tangible benefits for society.

Data mining is the process of discovering, extracting and analysing of meaningful patterns, structure, models, and rules from large quantities of data [10]. The process (appendix A) is automatic or semi-automatic, with interactive and iterative steps such as problem and data understanding, data selection, data preprocessing and cleaning, data transformation, incorporation of appropriate domain knowledge to select data mining task and algorithm, application of data mining algorithm(s), and knowledge interpretation and evaluation [69]. The last step is either the refinement by modifications or the consolidation of discovered knowledge [53]. Discovered patterns, or insights, should be statistically reliable, not known previously, and actionable [42, 128].

The data mining field spans several research areas [18] with stunning progress over the last decade. Database theories and tools provide the necessary infrastructure to store, access and manipulate data. Artificial intelligence research such as machine learning and neural networks is concerned with inferring models and extracting patterns from data. Data visualization examines methods to easily convey a summary and interpretation of the information gathered. Statistics is used to support and negate hypotheses on collected data and control the chances and risks that must be considered upon making generalisations. Distributed data mining deals with the problem of learning useful new information from large and inherently distributed databases where multiple models have to be combined.

The most common goal of business data mining applications is to predict customer behaviour. However this can be easily tailored to meet the objective of detecting and preventing criminal activity. It is almost impossible for perpetrators to exist in this modern era without leaving behind a trail of digital transactions in databases and networks [78].

Therefore, data mining in fraud detection is about systematically examining, in detail, hundreds of possible data attributes from such diverse sources as law enforcement, industry, government, and private data provider databases. It is also about building upon the findings, results and solutions provided by the database, machine learning, neural networks, data visualisation, statistics, and distributed data mining communities, to predict and deter illegitimate ac-

tivity.

The term fraud here refers to the abuse of a profit organisations system without necessarily leading to direct legal consequences. In a competitive environment, fraud can become a business critical problem if it is very prevalent and if the prevention procedures are not fail-safe. Fraud detection, being part of the overall fraud control, automates and helps reduce the manual parts of a screening/checking process. This area has become one of the most established industry/government data mining applications.

It is impossible to be absolutely certain about the legitimacy of and intention behind an application or transaction. Given the reality, the best cost effective option is to tease out possible evidences of fraud from the available data using mathematical algorithms.

Fraud takes many diverse forms, and is extremely costly to society. It can be classified into three main types, namely, against organisations, government and individuals (see appendix B). This thesis focuses on fraud against organisations.

Fraud detection poses some technical and practical problems for data mining; the most significant technical problem is due to limitations, or poor quality, in the data itself. The data is usually collected as a by-product of other tasks rather than for the purpose of fraud detection. Although one form of data collection standard for fraud detection has been introduced, not all data attributes are relevant for producing accurate predictions and some attribute values are likely to have data errors.

Another crucial technical dilemma is due to the highly skewed data in fraud detection. Typically there are many more legitimate than fraudulent examples. This means that by predicting all examples to be legal, a very high success rate is achieved without detecting any fraud. Another negative consequence of skewed data is the higher chances of overfitting the data. Overfitting occurs when models high accuracy arises from fitting patterns in the training set that are not statistically reliable and not available in the score set [42].

Another major technical problem involves finding the best ways to make predictions more understandable to data analysts.

The most important practical problem is a lack of domain knowledge, or prior knowledge, which reveals information such as the important attributes, the likely relationships and the known patterns. With some of the domain knowledge described in this and the following paragraph, the search time for using the data mining process can be reduced. Basically, fraud detection involves discovering three profiles of fraud offenders, each with constantly evolving

modus operandi. Average offenders can be of any gender or socio-economic group and they commit fraud when there is opportunity, sudden temptation, or when suffering from financial hardship. Criminal offenders are usually males and criminal records. Organised crime offenders are career criminals who are part of organised groups which are prepared to contribute considerable amount of time, effort and resources to perpetuate major and complex fraud.

Using learning algorithms in data mining to recognise a great variety of fraud scenarios over time is a difficult undertaking. Fraud committed by average offenders is known as soft fraud, which is the hardest to mitigate because the investigative cost for each suspected incident is usually greater than the cost of the fraud. Fraud perpetuated by the criminal and organised crime offenders is termed hard fraud and it circumvents anti-fraud measures and approximates many legal forms.

The next practical problem includes assessing the potential for significant impact of using data mining in fraud detection. Success cannot be defined in terms of predictive accuracy because of the skewed data.

Fraud detection represents a challenging issue in several application scenarios, and the automatic discovery of fraudulent behaviour is a very important task with great impact in many real-life situations. In this context, the Value Added Tax (VAT) fraud detection scenario is witnessing an increasing interest both for its practical and theoretical issues. Like any tax, the VAT is open to fraud and evasion. There are several ways in which it can be abused, e.g. by underdeclaring sales or overdeclaring purchases. However, opportunities and incentives to fraud are provided by the credit mechanism which characterizes VAT: tax charged by a seller is available to the buyer as a credit against their liability on their own sales and, if in excess of the output tax due, refunded to them. Thus, fraudulent claims for credit and refunds are an extensive and problematic issue in fiscal fraud detection. Under this perspective, the capability to provide a mathematical modelling methodology capable of producing a predictive analysis tool is of great significance. The tool should be able to identify the tax payers with the highest probability of being VAT defrauders, in order to support the activity of planning and performing effective fiscal audits.

There are several issues that make the problem difficult to address. First of all, the auditing capability is limited in each government agency: in Italy, for example, audited data available are only 0,004% of the overall population of taxpayers who file a VAT refund request. This restriction inevitably raises a sample selection bias: while auditing is the only way to produce a training set upon which to devise models, auditors focus only upon subjects which are particularly suspicious according to some clues. As a consequence, the number

of positive subjects (individuals which are actually defrauders) is much larger than the number of negative (i.e., non-defrauders) subjects. This implies that, despite the number of fraudulent individuals is far smaller than those of non-fraudulent individuals in the overall population, this proportion is reversed in the training set.

The limited auditing capability of a generic Revenue Agency poses severe constraints also in the design of the scoring system: auditing is a time-consuming task, involving several investigation and legal steps, which ultimately require a full-time employ of human resources. As a consequence, the scoring system should only concentrate on a user-defined fixed number of individuals (representing the auditing capability of the agency), with high fraudulent likelihood and with a minimum false positive rate. The DIVA project, that was report in section 3.2, tries to tackle the VAT Fraud Detection issue raised by the credit mechanism via the adoption of data mining techniques. The project involved computer science researchers, as well as experts from the Italian Revenue Agency and IT professional with expertise in managing the tax information system on behalf of the Italian Tax Administration.

The main objective of this study is to survey and evaluate methods and techniques to solve the common fraud detection problems previously outlined. Exactly, we want to individuate a methodology able to overcome the main problems of fraud detection for the VAT context and then propose a technique that combines the best methods considered to be effective for environments featured by the same problems of fraud detection, called "imprecise environments."

The fraud detection is an extremely difficult task mainly for the nature of the data, affected by bias, which makes it difficult to extract a suitable training set. For this reason, novel approaches have been developed to improve the accuracy of classification in imprecise (multi-class) learning environments, which are challenging domains wherein cases and classes of primary interest for the learning task are rare. The main thesis contributions are summarized below:

- To improve the predictive accuracy of models for the context of fraud detection it was introduced a new methodology to individuate fraudsters that aims to maximize the "quality" of those selected through the use of rule-based systems and ensemble methods. This technique is an ensemble method capable of combining the best of several rule-based baseline classification tools, each of them capable of addressing a specific problem among the ones described above. The idea of the approach is to progressively learn a set of rules until all the above requirements are met.

- In the real world it's difficult to choose a threshold, in a continuos range, to separate fruadsters more interesting than others. A new technique, is proposed for learning a model that deals with continuous values of exceptionality. Specically, given some training objects associated with a continuous attribute $\mathcal{F}$, it induces a rule-based model for the identification of those objects likely to score the maximum values for $\mathcal{F}$.

- A new framework is proposed that works in a more general area of data classification: imprecise data sets (noise), low occurrence of some cases of interest and low class separability. The framework introduces a hierarchical approach with two levels: at the top level, there is an associative classifier, which has a global view of data; at the lower level, there are a series of probabilistic models that have a local view of data, in particular one model for each rule of the associative classifier. The goal is to improve the performance, over the minority classes, of the associative classifier, combining it with the probabilistic models.

The rest of this dissertation is organized as follows:

Chapter 2 contains existing fraud detection methods and techniques, the new crime detection method and the evalution measures used in this enviroment.

Chapter 3 introduces a methodology, called *Sniper*, to predict, with high precision, fraudolent behaviours in VAT context;

Chapter 4 describes a technique to identify exceptional objects, in other words taken a set of objects ranked according to a continuos function $\mathcal{F}$this method lets to identify the objects with the higher values of $\mathcal{F}$;

Chapter 5 formalizes a hierarchical classification framework for discriminating rare classes in imprecise domains;

Chapter 6 concludes with the summary of the research, recommendations for the research problems, and possible directions for future research.

2

Fraud Detection: Methods and Techniques

2.1 Motivation

Studies have shown that detecting clusters of crime incidents [83] and finding possible cause/effect relations with association rule mining [43], are important to criminal analysis. Yet, the classification techniques have also proven to be highly effective in fraud detection [23, 59] and can be used to predict future crime data and to provide a better understanding of present crime data. Fraud detection involves identifying fraud as quickly as possible once it has been perpetrated. Fraud detection methods are continuously developed to defend criminals in adapting to their strategies. The development of new fraud detection methods is made more difficult due to the severe limitation of the exchange of ideas in fraud detection. Data sets are not made available and results are often not disclosed to the public. The fraud cases have to be detected from the available huge data sets such as the logged data and user behavior. Moreover, fraud detection data being highly skewed or imbalanced is the norm. Usually there are many more legitimate than fraudulent examples. This means that by predicting all instances to be legal, a very high success rate is achieved without detecting any fraud.

There can be two typical ways to proceed when faced with this problem. The first approach is to apply different algorithms (meta-learning). Each algorithm has its unique strengths, so that it may perform better on particular data instances than the rest [124]. The second approach is to manipulate the class distribution (sampling). The minority class training examples can be increased in proportion to the majority class in order to raise the chances of correct predictions by the algorithm(s). Most of the published work on improving the performance of standard classifiers on skewed data usually involves using the same algorithm(s). For example, the work on cost sensitive learning [39,90] aimed at reducing total cost, and sampling approaches [25,39] to favour the minority class are usually demonstrated with decision tree algorithms and/or naive Bayes.

One related problem caused by skewed data includes measuring the performance of the classifiers. Success cannot be defined in terms of predictive accuracy because the minority class in the skewed data usually has a significantly higher cost. Recent work on skewed data sets was evaluated using better performance metrics such as Area Under Curve (AUC) [25, 82], cost curves [41], and Receiver Operating Characteristic (ROC) analysis [76].

At present, fraud detection has been implemented by a number of methods such as data mining, statistics, and artificial intelligence. Fraud is discovered from anomalies in data and patterns. This chapter examines four major methods commonly used, and their corresponding techniques and algorithms, and introduces the main measures used to evaluate the results in this enviroment.

2.2 Overview

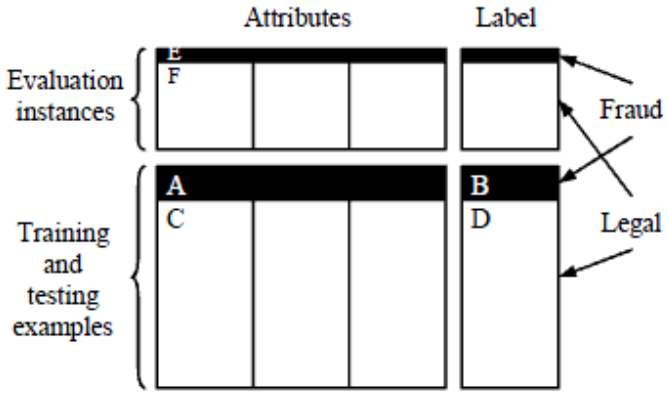


Fig. 2.1: Structured diagram of the possible data for analysis. Data mining approaches can utilise training/testing data with labels, only legal examples, and no labels to predict/describe the evaluation data.

The figure 2.1 shows that many existing fraud detection systems typically operate by adding fraudulent claims/applications/transactions/accounts/sequences (A) to black lists to match for likely frauds in the new instances (E). Some use hard-coded rules which each transaction should meet such as matching addresses and phone numbers, and price and amount limits [102].

An interesting idea borrowed from spam [51] is to understand the temporal nature of fraud in the black lists by tracking the frequency of terms and

category of terms (style or strategy of fraudster) found in the attributes of fraudulent examples over time. Below outlines the complex nature of data used for fraud detection in general [50, 51]:

- Volume of both fraud and legal classes will fluctuate independently of each other; therefore class distributions (proportion of illegitimate examples to legitimate examples) will change over time.
- Multiple styles of fraud can happen at around the same time. Each style can have a regular, occasional, seasonal, or onceoff temporal characteristic.
- Legal characteristics/behaviour can change over time.
- Within the near future after uncovering the current *modus operandi* of professional fraudsters, these same fraudsters will continually supply new or modified styles of fraud until the detection systems start generating false negatives again.

With reference to figure 2.1, the common data mining approaches to determine the most suspicious examples from the incoming data stream (evaluation data) are:

1. Labelled training data (A + B + C + D) can be processed by single *supervised* algorithms (section 2.3). A better suggestion is to employ hybrids such as multiple supervised algorithms (section 2.4.1), or both supervised and unsupervised algorithms (section 2.4.2) to output suspicion scores, rules and/or visual anomalies on evaluation data.
2. All known legal claims/applications/transactions/accounts/ sequences (C) should be used processed by *semi-supervised* algorithms to detect significant anomalies from consistent normal behaviour (section 2.5).
   However, there are many criticisms with using labelled data to detect fraud:
   - In an operational event-driven environment, the efficiency of processing is critical.
   - The length of time needed to flag examples as fraudulent will be the same amount of time the new fraud types will go unnoticed.
   - The class labels of the training data can be incorrect and subject to sample selectivity bias [58].

- They can be quite expensive and difficult to obtain [16].
- Staffs have to manually label each example and this has the potential of breaching privacy particularly if the data contains identity and personal information.
- [40] recommend the use of unlabelled data because the fraudster will try to make fraud and legal classes hard to distinguish.

Therefore it is necessary to:

3. Combine training data (the class labels are not required here) with evaluation data (A + C + E + F). These should be processed by single or multiple *unsupervised* algorithms to output suspicion scores, rules and/or visual anomalies on evaluation data (section 2.6).

2.3 Supervised Approaches on Labelled Data

Predictive supervised algorithms examine all previous labelled transactions to mathematically determine how a standard fraudulent transaction looks like by assigning a risk score [102]. Neural networks are popular and support vector machines (SVMs) have been applied. [36] used a three-layer, feed-forward Radial Basis Function (RBF) neural network with only two training passes needed to produce a fraud score in every two hours for new credit card transactions. [6] used a multi-layer neural network with exponential trace memory to handle temporal dependencies in synthetic Video-on-Demand log data. [126] propose fuzzy neural networks on parallel machines to speed up rule production for customer-specific credit card fraud detection. [65] proposes SVM ensembles with either bagging and boosting with aggregation methods for telecommunications subscription fraud.

The neural network and Bayesian network comparison study [77] uses the STAGE algorithm for Bayesian networks and backpropagation algorithm for neural networks in credit transactional fraud detection. Comparative results show that Bayesian networks were more accurate and much faster to train, but Bayesian networks are slower when applied to new instances.

[45] developed Bayesian network models in four stages with two parameters. They argue that regression, nearest-neighbour, and neural networks are too slow and decision trees have difficulties with certain discrete variables. The model with most variables and with some dependencies performed best for their telecommunications uncollectible debt data.

[109] applies the weight of evidence formulation of AdaBoosted naive Bayes (boosted fully independent Bayesian network) scoring. This allows the computing of the relative importance (weight) for individual components of suspicion and displaying the aggregation of evidence pro and contra fraud as a balance of evidence which is governed by a simple additivity principle. Compared to unboosted and boosted naive Bayes, the framework showed slightly better accuracy and AUC but clearly improved on the cross entropy and Brier scores. It is also readily accessible and naturally interpretable decision support and allows for flexible human expert interaction and tuning on an automobile insurance dataset.

Decision trees, rule induction, and case-based reasoning have also been used. [48] introduced systematic data selection to mine concept-drifting, possibly insufficient, data streams. The paper proposed a framework to select the optimal model from four different models (based on old data chunk only, new data chunk only, new data chunk with selected old data, and old and new data chunks). The selected old data is the examples which both optimal models at the consecutive time steps predict correctly. The crossvalidated decision tree ensemble is consistently better than all other decision tree classifiers and weighted averaging ensembles under all concept-drifting data chunk sizes, especially when the new data chunk size of the credit card transactions are small. With the same credit card data as [48], [112] demonstrates a pruned classifier C4.5 ensemble which is derived by weighting each base classifier according to its expected benefits and then averaging their outputs. The authors show that the ensemble will most likely perform better than a single classifier which uses exponential weighted average to emphasise more influence on recent data.

[97] presents a two-stage rules-based fraud detection system which first involves generating rules using a modified C4.5 algorithm. Next, it involves sorting rules based on accuracy of customer level rules, and selecting rules based on coverage of fraud of customer rules and difference between behavioural level rules. It was applied to a telecommunications subscription fraud. [13] used boosted C5.0 algorithm on tax declarations of companies. [100] applied a variant of C4.5 for customs fraud detection.

Case-based reasoning (CBR) was used by [121] to analyse the hardest cases which have been misclassified by existing methods and techniques. Retrieval was performed by thresholded nearest neighbour matching. Diagnosis utilised multiple selection criteria (probabilistic curve, best match, negative selection, density selection, and default) and resolution strategies (sequential resolution-default, best guess, and combined confidence) which analysed the retrieved cases. The authors claimed that CBR had 20% higher true positive and true negative rates than common algorithms on credit applications.

Statistical modelling such as regression has been extensively utilised. [54] use

least squares regression and stepwise selection of predictors to show that standard statistical methods are competitive. Their version of fully automatic stepwise regression has three useful modifications: firstly, organises calculations to accommodate interactions; secondly, exploits modern decision-theoretic criteria to choose predictors; thirdly, conservatively estimate p-values to handle sparse data and a binary response before calibrating regression predictions. If cost of false negative is much higher than a false positive, their regression model obtained significantly lesser misclassification costs than C4.5 for telecommunications bankruptcy prediction.

[46] chooses the best indicators (attributes) of fraud by first querying domain experts, second calculating conditional probabilities of fraud for each indicator and third Probit regressions to determine most significant indicators. The authors also use Prohit regressions to predict fraud and adjusts the threshold to suit company fraud policy on automobile property damages. [75] compares a multinomial logit model (MNL) and nested multinomial logit model (NMNL) on a multiclass classification problem. Both models provide estimated conditional probabilities for the three classes but NMNL uses the two step estimation for its nested choice decision tree. It was applied to automobile insurance data. [79] described least-squares stepwise regression analysis for anomaly detection on aggregated employees applications data.

Other techniques include expert systems, association rules, and genetic programming. Expert systems have been applied to insurance fraud. [37] have implemented an actual five-layer expert system in which expert knowledge is integrated with statistical information assessment to identify medical insurance fraud. [99]), [47] and [111] have experimented on fuzzy expert systems. [105] applied an expert system to management fraud. [27] introduce a Fraud Patterns Mining (FPM) algorithm, modified from Apriori, to mine a common format for fraud-only credit card data. [8] uses genetic programming with fuzzy logic to create rules for classifying data. This system was tested on real home insurance claims [8] and credit card transaction data [9]. None of these papers on expert systems, association rules, and genetic programming provide any direct comparisons with the many other available methods and techniques.

The above supervised algorithms are conventional learning techniques which can only process structured data from single 1- to-1 data tables. Further research using labelled data in fraud detection can benefit from applying relational learning approaches such as Inductive Logic Programming (ILP) [80] and simple homophily-based classifiers (Provost et al, 2003) on relational databases. [86] also present novel target-dependent aggregation methods for converting the relational learning problem into a conventional one.

2.4 Hybrid Approaches with Labelled Data

2.4.1 Supervised Hybrids

Popular supervised algorithms such as neural networks, Bayesian networks, and decision trees have been combined or applied in a sequential fashion to improve results. [23] utilises naive Bayes, C4.5, CART, and RIPPER as base classifiers and stacking to combine them. They also examine bridging incompatible data sets from different companies and the pruning of base classifiers. The results indicate high cost savings and better efficiency on credit card transactions. [88] proposes backpropagation neural networks, naive Bayes, and C4.5 as base classifiers on data partitions derived from minority oversampling with replacement. Its originality lies in the use of a single meta-classifier (stacking) to choose the best base classifiers, and then combine these base classifiers predictions (bagging) to produce the best cost savings on automobile insurance claims.

[85] recommends a rule generator to refine the weights of the Bayesian network. [106] propose a decision tree to partition the input space, tanh as a weighting function to generate fraud density, and subsequently a backpropagation neural network to generate a weighted suspicion score on credit card transactions.

Also, [59] propose genetic algorithms to determine optimal weights of the attributes, followed by k-nearest neighbour algorithm to classify the general practitioner data. They claim significantly better results than without feature weights and when compared to CBR.

2.4.2 Supervised/Unsupervised Hybrids

There is extensive work on labelled data using both supervised and unsupervised algorithms in telecommunications fraud detection. [33] propose the use of signatures (telecommunication account summaries) which are updated daily (time-driven). Fraudulent signatures are added to the training set and processed by supervised algorithms such as atree, slipper, and model-averaged regression. The authors remark that fraudulent toll-free numbers tend to have extensive late night activity and long call durations. Cortes and Pregibon (2001) use signatures assumed to be legitimate to detect significant changes in calling behaviour. Association rules is used to discover interesting country combinations and temporal information from the previous month. A graph-theoretic method [34] is used to visually detect communities of interest of fraudulent international call accounts (see section 2.6). [20] assign an averaged suspicion score to each call (event-driven) based on its similarity to fraudulent signatures and dissimilarity to its accounts normal signature. Calls with low

scores are used to update the signature and recent calls are weighted more heavily than earlier ones in the signature.

[50] present fraud rule generation from each cloned phone accounts labelled data and rule selection to cover most accounts. Each selected fraud rule is applied in the form of monitors (number and duration of calls) to the daily legitimate usage of each account to find anomalies. The selected monitors output and labels on an accounts previous daily behaviour are used as training data for a simple Linear Threshold Unit. An alarm will be raised on that account if the suspicion score on the next evaluation day exceeds its threshold. In terms of cost savings and accuracy, this method performed better than other methods such as expert systems, classifiers trained without account context, high usage, collision detection, velocity checking, and dialled digit analysis on detecting telecommunications superimposed fraud.

Two studies on telecommunications [21] data show that supervised approaches achieve better results than unsupervised ones. With AUC as the performance measure, [101] show that supervised neural network and rule induction algorithms outperform two forms of unsupervised neural networks which identify differences between short-term and long-term statistical account behaviour profiles. The best results are from a hybrid model which combines these four techniques using logistic regression. Using true positive rate with no false positives as the performance measure, [107] claim that supervised neural networks and Bayesian networks on labelled data achieve significantly better outcomes than unsupervised techniques such as Gaussian mixture models on each non-fraud user to detect anomalous phone calls.

Unsupervised approaches have been used to segment the insurance data into clusters for supervised approaches. [28] applies a three step process: k-means for cluster detection, C4.5 for decision tree rule induction, and domain knowledge, statistical summaries and visualisation tools for rule evaluation. [122] use a genetic algorithm, instead of C4.5, to generate rules and to allow the domain user, such as a fraud specialist, to explore the rules and to allow them to evolve accordingly on medical insurance claims. [95] present a similar methodology utilising the Self Organising Maps (SOM) for cluster detection before backpropagation neural networks in automobile injury claims. [35] uses an unsupervised neural network followed by a neuro-fuzzy classification system to monitor medical providers claims.

Unconventional hybrids include the use of backpropagation neural networks, followed by SOMs to analyse the classification results on medical providers claims [98] and RBF neural networks to check the results of association rules for credit card transactions [14].

2.5 Semi-supervised Approaches with Only Legal Data

[66] implements a novel fraud detection method in five steps: First, generate rules randomly using association rules algorithm Apriori and increase diversity by a calender schema; second, apply rules on known legitimate transaction database, discard any rule which matches this data; third, use remaining rules to monitor actual system, discard any rule which detects no anomalies; fourth, replicate any rule which detects anomalies by adding tiny random mutations; and fifth, retain the successful rules. This system has been and currently being tested for internal fraud by employees within the retail transaction processing system.

[81] use profiling at call, daily, and overall levels of normal behaviour from each telecommunications account. The common daily profiles are extracted using a clustering algorithm with cumulative distribution distance function. An alert is raised if the daily profiles call duration, destination, and quantity exceed the threshold and standard deviation of the overall profile. [2] experiment with auto-associative neural networks (one hidden layer and the same number of input and output neurons) on each credit card accounts legal transactions. [67] proposes similarity trees (decision trees with Boolean logic functions) to profile each legitimate customers behaviour to detect deviations from the norm and cluster analysis to segregate each legitimate customers credit card transactions.

2.6 Unsupervised Approaches with Unlabelled Data

Link analysis and graph mining are hot research topics in antiterrorism, law enforcement, and other security areas, but these techniques seem to be relatively under-rated in fraud detection research. A white paper [84] describes how the emergent group algorithm is used to form groups of tightly connected data and how it led to the capture of an actual elusive fraudster by visually analysing twelve months worth of insurance claims. There is a brief application description of a visual telecommunications fraud detection system [35] which flexibly encodes data using colour, position, size and other visual characteristics with multiple different views and levels. The intuition is to combine human detection with machine computation.

[34] examines temporal evolution of large dynamic graphs for telecommunications fraud detection. Each graph is made up of subgraphs called Communities Of Interest (COI). To overcome instability of using just the current graph, and storage and weightage problems of using all graphs at all time steps; the authors used the exponential weighted average approach to update subgraphs daily. By linking mobile phone accounts using call quantity and durations to form COIs, the authors confirm two distinctive characteristics of fraudsters.

First, fraudulent phone accounts are linked - fraudsters call each other or the same phone numbers. Second, fraudulent call behaviour from flagged frauds are reflected in some new phone accounts - fraudsters retaliate with application fraud/identity crime after being detected. [34] states their contribution to dynamic graph research in the areas of scale, speed, dynamic updating, condensed representation of the graph, and measure direct interaction between nodes.

Some forms of unsupervised neural networks have been applied. [40] creates a non-linear discriminant analysis algorithm which do not need labels. It minimises the ratio of the determinants of the within and between class variances of weight projections. There is no history on each credit card accounts past transactions, so all transactions have to be segregated into different geographical locations. The authors explained that the installed detection system has low false positive rates, high cost savings, and high computational efficiency. [17] use a recurrent neural network to form short-term and long-term statistical account behaviour profiles. Hellinger distance is used to compare the two probability distributions and give a suspicion score on telecommunications toll tickets.

In addition to cluster analysis (section 2.4.2), unsupervised approaches such as outlier detection, spike detection, and other forms of scoring have been applied. [127] demonstrated the unsupervised SmartSifter algorithm which can handle both categorical and continuous variables, and detect statistical outliers using Hellinger distance, on medical insurance data.

[17] recommend Peer Group Analysis to monitor inter-account behaviour over time. It compares the cumulative mean weekly amount between a target account and other similar accounts (peer group) at subsequent time points. The distance metric/suspicion score is a t-statistic which determines the standardised distance from the centroid of the peer group. The time window to calculate peer group is thirteen weeks and future time window is four weeks on credit card accounts. [12] also suggest Break Point Analysis to monitor intraaccount behaviour over time. It detects rapid spending or sharp increases in weekly spending within a single account. Accounts are ranked by the t-test. The fixed-length moving transaction window contains twenty-four transactions: first twenty for training and next four for evaluation on credit card accounts.

[16] recommends Principal Component Analysis of RIDIT scores for rank-ordered categorical attributes on automobile insurance data.

[62] present an experimental real-time fraud detection system based on a Hidden Markov Model (HMM).

2.7 Performance Measures

Most fraud departments place monetary value on predictions to maximise cost savings/profit and according to their policies. They can either define explicit cost [23, 50, 88, 112] or benefit models [48, 112].

[19] suggests giving a score for an instance by determining the similarity of it to known fraud examples (fraud styles) divided by the dissimilarity of it to known legal examples (legitimate telecommunications account).

Most of the fraud detection studies using supervised algorithms since 2001 have abandoned measurements such as true positive rate (correctly detected fraud divided by actual fraud) and accuracy at a chosen threshold (number of instances predicted correctly, divided by the total number of instances). In fraud detection, misclassification costs (false positive and false negative error costs) are unequal, uncertain, can differ from example to example, and can change over time. In fraud detection, a false negative error is usually more costly than a false positive error. Regrettably, some studies on credit card transactional fraud [26] and telecommunications superimposed fraud [66] still aim to only maximise accuracy. Some use Receiver Operating Characteristic (ROC) analysis (true positive rate versus false positive rate).

Apart from [110], no other fraud detection study on supervised algorithms has sought to maximise Area under the Receiver Operating Curve (AUC) and minimise cross entropy (CXE). AUC measures how many times the instances have to be swapped with their neighbours when sorting data by predicted scores; and CXE measures how close predicted scores are to target scores. In addition, [110] and [54] seek to minimise Brier score (mean squared error of predictions). Caruana and Niculescu-Mizil (2004) argues that the most effective way to assess supervised algorithms is to use one metric from threshold, ordering, and probability metrics; and they justify using the average of mean squared error, accuracy, and AUC. [52] recommend Activity Monitoring Operating Characteristic (AMOC) (average score versus false alarm rate) suited for timely credit transactional and telecommunications superimposition fraud detection.

For semi-supervised approaches such as anomaly detection, [70] propose entropy, conditional entropy, relative conditional entropy, information gain, and information cost. For unsupervised algorithms, [127] used the Hellinger and logarithmic scores to find statistical outliers for insurance; [17] and [101] employed Hellinger score to determine the difference between short-term and longterm profiles for the telecommunications account. [12] recommends the t-statistic as a score to compute the standardised distance of the target account with centroid of the peer group; and also to detect large spending changes within accounts.

Other important considerations include how fast the frauds can be detected (detection time/time to alarm), how many styles/types of fraud detected, whether the detection was done in online/real time (event-driven) or batch mode (time-driven) [36].

There are problem-domain specific criteria in insurance fraud detection. To evaluate automated insurance fraud detection, some domain expert comparisons and involvement have been described. [111] claimed that their algorithm performed marginally better than the experienced auditors. [16] and [47] summed up their performance as being consistent with the human experts and their regression scores. [46] stated that both automated and manual methods are complementary. [122] supports the role of the fraud specialist to explore and evolve rules. [84] reports visual analysis of insurance claims by the user helped discover the fraudster.

2.8 Critique of Methods and Techniques

Each technique is intrinsically different from the other, according to the evaluation criteria, and has its own strengths and weaknesses. *Interpretability* refers to how much a domain expert or non-technical person can understand each of the model predictions through visualisations or rules. *Effectiveness* highlights the overall predictive accuracy and performance of the each technique. *Robustness* assesses the ability to make correct predictions given noisy data or data with missing values. *Scalability* refers to the capability to construct a model efficiently given large amounts of data. *Speed* describes how effective it is in terms of how fast a technique searches for patterns that make up the model.

For example BBNs could be used for *scalability* and *speed*, decision trees for *interpretability*, and ANNs for its *effectiveness* and *robustness*. By abstracting from the peculiarities of each of the above techniques, we can generally affirm:

- In most scenarios of real-world fraud detection, the choice of data mining techniques is more dependent on the practical issues of operational requirements, resource constraints, and management commitment towards reduction of fraud than the technical issues poised by the data.

- There is too much emphasis by research on complex, nonlinear supervised algorithms such as neural networks and support vector machines. In the long term, less complex and faster algorithms such as naive Bayes [109]

and logistic regression [72] will produce equal, if not better results, on population-drifting, concept-drifting, adversarial-ridden data. If the incoming data stream has to be processed immediately in an event-driven system or labels are not readily available, then semisupervised and unsupervised approaches are the only data mining options.

- Other related data mining techniques used in this enviroment and covered by survey papers and bibliographies include outlier detection [60], skewed /unbalanced /rare classes [60], sampling [38], cost sensitive learning, stream mining, graph mining [114], and scalability [91].

3

SNIPER: a methodology for Fiscal Fraud Detection

3.1 SNIPER technique

Planning adequate audit strategies is a key success factor in a posteriori fraud detection applications, such as in fiscal and insurance domains, where audits are intended to detect fraudulent behavior. SNIPER is an auditing methodology based on a rule-based system, which is capable of trading among conflicting issues, such as maximizing audit benefits, minimizing false positive audit predictions, or deterring probable upcoming frauds. In this chapter it is described the experience made on the Value Added Tax (VAT) fraud detection scenario and the preliminaries results obtained using SNIPER approach.

3.2 Application Context

The objective of the DIVA project, introduced in section **??**, was to design a predictive analysis tool able to identify the tax payers with the highest probability of being VAT defrauders to the aim of supporting the activity of planning and performing effective fiscal audits. The construction of the model is based on historical VAT declaration records labeled with the outcome of the audit performed by the Agency. The domain of the DIVA project is particularly challenging both from a scientific and a practical point of view. First of all, audited data available represent a very small fraction (about 0.15%) of the overall population of taxpayers requesting a VAT refund. This resource-aware restriction inevitably raises a sample selection bias. Indeed, auditing is the only way to produce a training set, and auditors focus only upon subjects which are particularly suspicious according to some clues. As a consequence, the number of positive subjects (individuals which are actually defrauders) is much larger than the number of negative (i.e., non-defrauders) subjects. This implies that, despite the number of fraudulent individuals is far smaller than those of non-fraudulent individuals in the overall population,

this proportion is reversed in the training set. Since auditing is resource-consuming, the number of individuals reported as possible fraudsters is of high practical impact. Hence, a scoring system should primarily suggest subjects with a high fraudulent likelihood, while minimizing false positives. From a socio-economic point of view, it is preferable to adopt a rule based approach to modeling. Indeed, intelligible explanations about the reason why individuals are scored as fraudulent are by far more important than the simple scores associated with them, since they allow auditors to thoroughly investigate the behavioral mechanisms behind a fraud.

3.3 DIVA Overview

In this section we provide an overview of the experience we tackled and the related technique we propose. The section is intended to clarify the choices about the formal building raised up. The data coming from the governmental Revenue Agency is concerned with the VAT declarations of Italian business companies. In particular the experience is focused on the companies claiming a VAT refund. The data made available by the agency consisted of about 34 million VAT declarations spread over 5 years. Data contain general demographic information, like Zip of the registered office, start-up year and Legal status, plus specific information about VAT declarations, like Business Volume, Sales, Import, Export and the total amount of VAT Refund. As a result of a data understanding process conducted jointly with domain experts, we chose a total of 135 such features.

Out of the 34 million of declarations, we collected further information about 45,442 audited subjects. The results of auditing for such subjects are summarized in the further feature VAT refund fraud (the difference between the amount of VAT Refund claimed and the VAT Refund actually due). Thus, audited subjects can be roughly classified into defrauders (when *VAT refund fraud* ≥ 0) and non-defrauders (in the other case). The resulting labeled training set is extremely biased, consisting of 38,759 (85.29%) subjects belonging to the defrauder class, and 6,683 (14.71%) belonging to the non-defrauder class.

The situation is further exacerbated by the quest for a multi-purpose modeling methodology. Experts are interested in scoring individuals according to the following three main criteria.

- *Proficiency:* scoring and detection should rely not only on a binary decision boundary separating defrauders from non-defrauders. Better, higher fraud amounts make defrauders more significant. For example, detecting a defrauder whose fraud amounts to $1,000 is better than discovering a

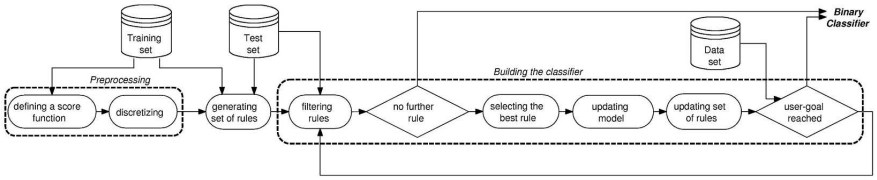


Fig. 3.1: Flowchart of the SNIPER technique

defrauder whose fraud amounts to $100.

- *Equity:* a weighting mechanism should leverage detection and scoring to include those cases where the amount of fraud is relevant related to their business volume. In practice, it should be avoided that individuals with low business volumes are never audited. For example, an individual whose fraud amounts to $1,000 and whose business volume amounts to $100,000 , is less interesting than an individual whose fraud amounts to $1,000 but the business volume amounts to $10,000.

- *Efficiency:* Since the focus is on refunds, scoring and detection should be sensitive to total/partial frauds. For example, a subject claiming an amount of VAT refund equal to 2,000 and entitled to 1,800 is less significant than a different subject claiming 200 and entitled to 0.

A further requirement is represented by the limited auditing capacity of the Revenue Agency: auditing is a very timeconsuming task, involving several investigation and legal steps which ultimately require a full-time employee. As a consequence, the scoring system should retrieve from the population a user-defined fixed number of individuals with high defrauder likelihood.
Sniper has been devised to accommodate all the above mentioned issues in a unified framework. The idea of the approach is to progressively learn a set of rules until all the above requirements are met. The approach is summarized in figure 3.1.

As a first step, a score function is computed which associates an individual with a value representing its degree of interestingness according to the proficiency, equity and efficiency parameters. Clear enough, this function is not known for the individuals in the whole population. Nevertheless, the *training set* of audited subjects allows the computation of such a function and its analytical evaluation over those known cases.

A discretization step is accomplished for the scoring function, thus associating a class label to each discretization level. This leads to the definition of a class containing the individuals scoring to the maximum value of the function.

Such a class is referred to in the following as *top class*.

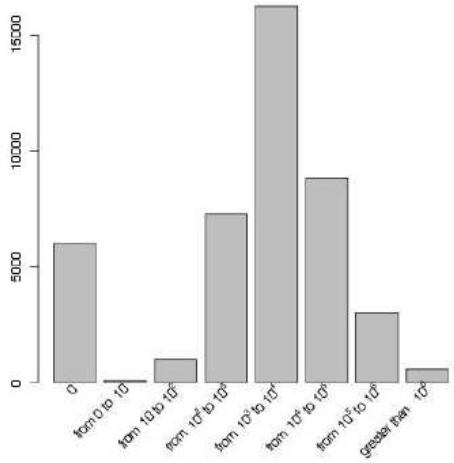
The main objective is hence to build a rule set able to identify individuals belonging to the top class, with two main objectives: (i) false positives should be minimized; (ii) the number of subjects should be as close as possible to a user-specified value. To this purpose, a set of classifiers is trained, where each classifier provides a set of rules. These sets are collected in a global set $\mathcal{R}$ after a filtering phase that removes rules not complying with a minimum quality criteria. The set of rules $\mathcal{R}$ taken as a whole is not, in general, the best according to the two objectives cited above, since (i) its accuracy (the percentage of subjects of the top class retrieved) can be too low, and (ii) the number of retrieved subjects can be too high. This will be better clarified in section 3.5.

The global set of rules $\mathcal{R}$ is employed as input in order to build a final binary classifier, consisting in the optimal subset of the rules in $\mathcal{R}$, according to the two main quality criteria. Notice that the problem of finding the best subset is intractable, thus Sniper uses a greedy strategy. The latter consists into iteratively selecting the best rule, until the quality criteria are met.

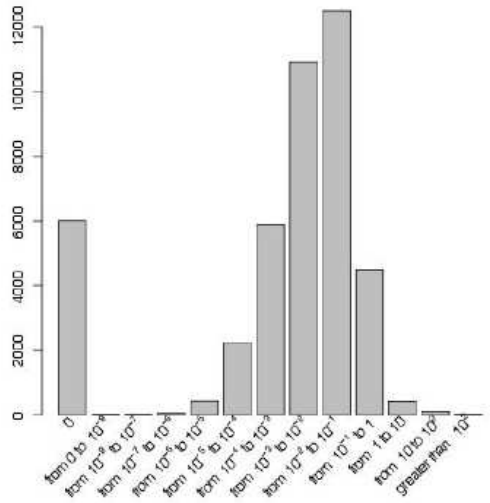
3.4 Modeling Multi-Purpose Objectives

As mentioned in the previous section, a primary task in VAT fraud is to formalize the notion of interestingness and exceptionalness of an individual. As already stated, auditing individuals is a very resource-consuming task and then it should be focused on those individuals which, among the defrauders, are the most interesting ones.

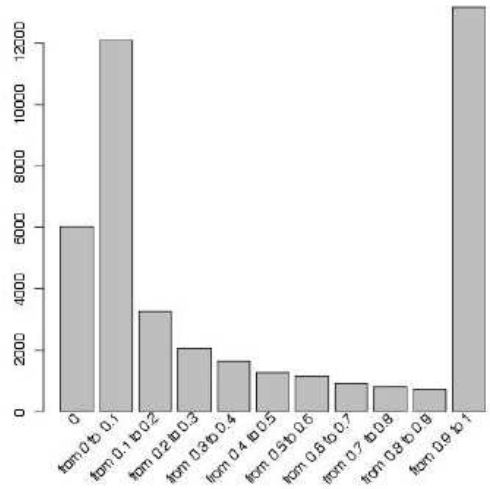
Within DIVA, this is formalized by means of a scoring function, which allows to associate a rank to the whole pupulation and thus to detect the *top-fraudulent* individuals. This approach is often preferable to a rough classification of the population into fraudulent and non-fraudulent individuals. Here, the notion of scoring function is different than fraud likelihood, i.e., the likelihood of an individual being defrauder. We are more interested in characterizing the severity of the committed fraud. This is done by devising a scoring function capable of associating a notion of severity. In a sense, this is a regression problem, since the scoring function ranges into an interval of continuous values. Of course, people unlikely to commit fraud scores to the lowest values. However, an individual can be associated with a high probability of committing a fraud, but his/her relevance is low, since the amount of fraud committed is low (e.g., less than 1000). Under this perspective, the score associated with such an individual should be low. Moreover, higher scoring



(a) Fraud amount



(b) Ratio between fraud amount and Business volume



(c) Ratio between fraud amount and VAT claimed

Fig. 3.2: Histograms describing the distributions of proficiency (3.2a), efficiency (3.2b) and equity (3.2c).

levels represent high relevance, but an individual may not necessarily represent a high likelihood of scoring to such levels.
Thus, our objective is to twofold:

- first, to devise a scoring function based on the notion of fraud severity;
- next, to associate, for each individual, the likelihood of committing a severe fraud (i.e., to scoring to the highest values of the devised scoring function).

Unfortunately, several facets contribute to the definition of severity. Consider, e.g., the histograms in figure 3.2. they represents respectively proficiency, equity and efficiency. However, while all the graphs show a base of almost 7000 individuals with low inclination to fraud, the distribution for the remaining subjects is substantially different. In particular, both proficiency and efficiency follow a lognormal distribution, whereas equity exhibits a mixture of two different behaviors, peaked at different degrees of severity.

In the above example, no single measure summarizes the notion of severity of fraud. Better, it could be modeled as a combination of the baseline functions: e.g., within figure 3.2, the individuals scoring to the highest values should lay into the intersection of the rightmost buckets. Other subjects can be selected according to specific user-defined criteria: e.g., we could be interested in individuals scoring high values in any of the three measures, or scoring in a similar way on all of them. We devise this multi-purpose strategy in two stages. In a first stage, several baseline scoring fuctions, also called first level function, can be defined, where each function is aimed at highlighting a specific aspect of the fraudulent behavior. In a second stage, baseline functions are combined according to specific business objectives, thus allowing to focus on the aspects of main interest and to better tune specific auditing criteria.

In general, many aspects play a role about the user notion of interesting individuals, therefore many parameters should be taken into account for estimating an individual as interesting one. The idea here pursued is to define, interacting with the user, a function (called *first-level function*) for each of such parameters; then to combine them using a *second-level function*, able to weight the different first level functions in order to match as best as possible the user needs; and finally to define a scoring function able to sort the individuals by their *interestingness*.

Baseline scoring functions can be used in combination with threshold values, in order to highlight exceptional values to be taken into account in the next stages. For a particular baseline function f, two threshold values σ_f^{hi} and σ_f^{lo} can be defined, which partition the sample into the clusters of individuals o such that $f(o) \leq \sigma_f^{lo}$, $\sigma_f^{lo} < f(o) \leq \sigma_f^{hi}$, and $f(o) > \sigma_f^{hi}$. An individual as-

suming a very high value on a baseline function f (higher than σ_f^{hi}) can be of interest for the user even if its score (valuated in the combination with other baselines) is not very high. Analogously, if an individual assumes a very low value on a first-level function (below σ^{lo}) and it is not outstanding (beyond σ^{hi}) in any of the baseline functions, then it is not interesting for the user even if its score is high.

Another advantage for the use of the baseline scoring functions is the possibility to mitigate the sample-selection bias arising from the non-random choice of the individuals to be audited. Consider again the distributions in figure 3.2. We call the functions corresponding to such distributions, f_{prof} (representing the total amount of fraud), f_{equ}, (the ratio between the total fraud and the business volume), and f_{eff} (the ratio between the total fraud amount and the total VAT refund declared). The sample is clearly biased towards fraudsters, most individuals exhibit values greater than 0. The adoption of a lower threshold σ_f^{lo}, however, allows to rebalance the situation to a more realistic distribution. The figure 3.3 reports the distribution of defrauders and non-defrauders subjects belonging to the sample set. The first histogram represents the distribution as partitioned by f_{prof} with threshold 0. Note that this corresponds to roughly classify as non-defrauders those subjects whose total fraud amount is 0 and as defrauders all the other subjects, denoted as f_{basic}. The other histograms represent the distribution as partitioned by f_{prof}, f_{equ}, and f_{eff}, respectively. Threshold were chosen by exeperts, who assigne the values $\sigma^{lo}_{f_{prof}} = 2{,}000$, $\sigma^{lo}_{f_{equ}} = 0.0025$, and $\sigma^{lo}_{f_{eff}} = 0.2$, respectively.

It is important to notice that a careful choice of the threshold values does not alter the significance of the training set. The figure 3.4 shows the retrieved fraud (i.e., the sum of the VAT refund fraud) associated with both the subjects identified as defrauders and as non-defrauders by each of the first level functions above considered.
The figure 3.3 highlights that the size of the set S_f of subjects identified as defrauders by the first level function f is strongly reduced with respect to the size of the set $S_{f_{basic}}$ of defrauders identified by f_{basic}. Nevertheless, as shown in figure 3.4 the retrieved fraud of S_f is almost similar to that in $S_{f_{basic}}$, thus confirming that the most interesting defrauders are those selected by the baseline functions.

First level functions play a major role in modeling local properties of fraudulent behavior. The role of a second level function is to combine such local properties into a global interestingness measure capable of summarizing them. We can formalize it as follows. Given k first level objective functions $f_1, \ldots, f_k$, a *second-level objective function* is a function $\mathcal{F}$, associating each individual of a population with a real number ranging in [0,1], by combining the values assumed by $f_1, \ldots, f_k$. The contribution of f_i can be also weighted in order

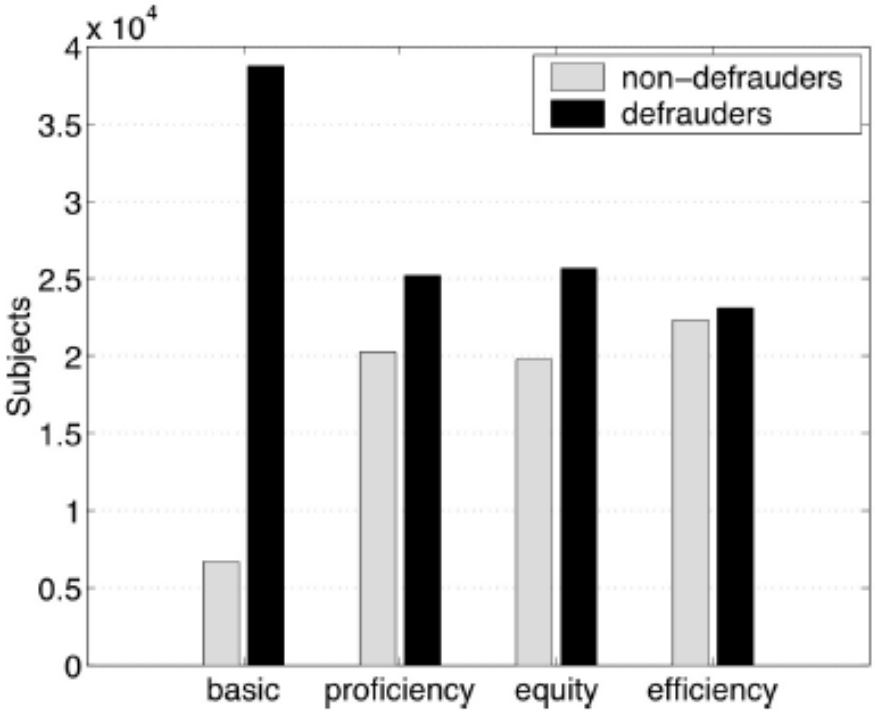


Fig. 3.3: Training set partitioning according to first-level functions

to tune its influence within $\mathcal{F}$.

The combination is made of two steps. A first preliminary step consists in harmonizing the values of the first-level functions, as they can be in different ranges and scales. Consider for example, the function f_{prof} and the function f_{equ}. The former represents the absolute value of the fraud amount, while the latter represents the ratio between the fraud amount and the business volume, thus ranging in $[0, 1]$. Directly combining them is clearly misleading as they refer to different unit measures.

Harmonization should also take care of rescaling values according to threshold values, in order to preserve homogeneity in comparisons. Consider for example two functions f_1 and f_2, both ranging in $[0, 1]$, whose thresholds are $\sigma_1^{lo} = 0.01$, $\sigma_1^{hi} = 0.1$, $\sigma_2^{lo} = 0.7$ and $\sigma_2^{hi} = 0.9$. If for an object o both $f_1(o)$ and $f_2(o)$ assume value 0.5, the semantic of such a value is inherently different, and a combination of such values without a proper adjustment would result into a misleading score.

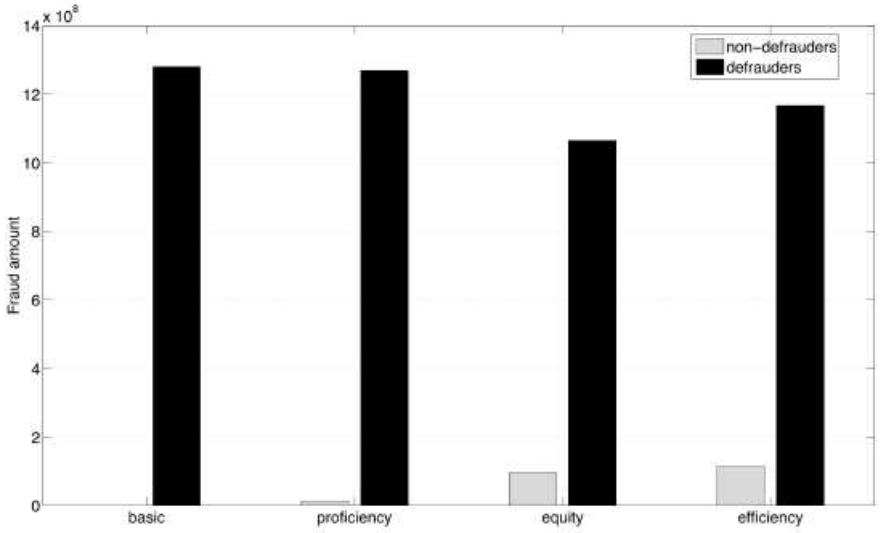


Fig. 3.4: Retrieved fraud within the partitioned dataset.

Within Sniper, harmonization is accomplished by means of a *normalizing* function $\mathcal{N} : \mathbb{R} \to [0, 1]$, associating each value assumed by a first level function with a value in the range $[0, 1]$. $\mathcal{N}$ can simultaneously account for the normalization concerning scales, ranges and thresholds.

Second-level functions can be directly derived by combining and weighting the normalized versions of the first-level functions. We considered two main combination functions:

$$\mathcal{F}_{\Pi}(o) = \prod_{i \in [1,k]} (\mathcal{N}(f_i(o)))^{p_i}$$

$$\mathcal{F}_{\Sigma}(o) = \sum_{i \in [1,k]} p_i \cdot \mathcal{N}(f_i(o)),$$

where p_i represents the weight associated with f_i. The $\mathcal{F}_{\Pi}$ function returns the weighted product of the f_i, whereas the $\mathcal{F}_{\Sigma}$ function returns the weighted sum of the f_i.

These two functions satisfy a different conceptual enforcement, but both of them have guaranteed good experimental results. Essentially, the former function is built by applying a sort of conjunctive operator to the single first-level functions; this fact causes that $\mathcal{F}_{\Pi}(o)$ assigns an higher value to those subjects having high values for each first level functions. The latter instead implements a disjunctive criteria, which associates a high value with those

subjects having an high value for one first level functions at least.

Thus, the $\mathcal{F}_{\Pi}$ function is more selective than $\mathcal{F}_{\Sigma}$ and therefore it could assign a low value to some interesting subjects, for instance, characterized by a low value for one first level function at least and a very high value for the other ones. Analogously, $\mathcal{F}_{\Sigma}$ suffers of the opposite problem, namely, it could assign an high value to those subjects having an high value for one first level function but low values for all the other ones.

The adequacy of a second level function for capturing the most prominent aspects of the first level functions can be appreciated in figure 3.5. Here, we show the cumulative gains obtained for decreasing values of the score function (equipped with $\mathcal{F}_{\Pi}$), relative to profitability, equity and efficiency. Notice that top individuals cumulate the largest gain in practically all the three parameters.

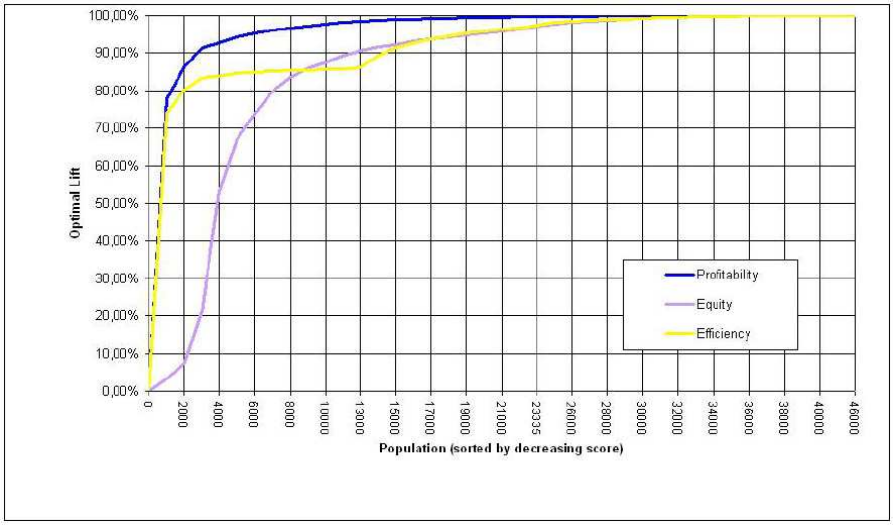


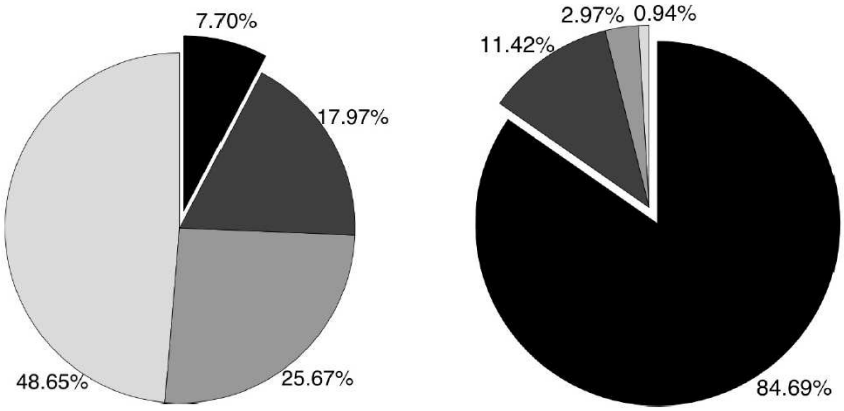
Fig. 3.5: Cumulative gains in proficiency, equity and efficiency related to the score function.

In the framework we are addressing, the goal is to retrieve from the population X individuals scoring the maximum value for the score function. It is more convenient to address this problem by discretizing the $\mathcal{F}$ function of interest. This allows us to gain more control on the prediction error, and to adopt well-known classification algorithms to our framework. When discretizing, we can split the scoring function into intervals, and the interval containing the highest values of the score function identifies the top class.

The width of the top class strongly influences the quantity and the quality of the individuals identified as members of the top class in the population. Hence, the discretization phase plays a very important role.

Discretization was accomplished by studying the distribution of the score functions. As a result, four main classes were detected. Figure 3.6a reports the effects of the employed discretization in partitioning the subjects. Specifically, from the lighter to the darker colored slice, the figure reports the percentage of subjects in classes 0, 1, 2 and 3, respectively. Conversely, Figure 3.6b reports the percentage of total amount of fraud made by the subjects of the different classes. It is worth pointing out that the subjects belonging to the top class, represent only 7.70% of the total number of audited subjects, but the total amount of fraud associated with them is 84.69% of the total fraud amount made by the whole set of audited subjects. This confirms the adequacy of the score function and the related discretization to our needs.

It is worth noticing also that different approaches can be exploited to discretization, based on clustering algorithms. In principle, the values of first and second level functions can be used to characterize groups. A cluster in that case would represent a set of individuals exhibting similar fraudulent behavior. The interested reader can find the details of the clustering approach to discretization in [7].



(a) Subject partitioning (b) Retrieved fraud

Fig. 3.6: Score function results

3.5 Building the classifier

The core of the Sniper technique is the extraction of a binary classifier able to identify X defrauders in the dataset, likely to be the most fraudulent individuals. As introduced in section 3.2, the Sniper technique trains a set of classifiers on the top class detected in the training set by the preprocessing phase. These classifiers are then merged into a single ruleset which is further processed.

Some preliminary notions follow. An *attribute* a is an identifier with an associated domain, called $Dom(a)$. Let $A = \{a_1, \ldots, a_m\}$ be a set of m attributes and C a special attribute called *class attribute*. An *object* o on A and C is a pair $(\mathbf{v}, c)$ where $\mathbf{v}$ is an m-ple $\langle v_1, \ldots, v_m \rangle$ of values belonging to the domain of $a_1, \ldots, a_m$ respectively, and c is the value of the class attribute, also called *the class of* o. A *dataset* D on is a multi-set of objects on A and $\mathcal{C}$. A *condition* on A is an expression of the form $a \in V$, where $a \in A$ and $V \subseteq Dom(a)$. The expression $a \notin V$ is a shortcut for the condition $a \in Dom(a) \setminus V$. An object o satisfies a conjunction $B = (a_1 \in V_1 \wedge \cdots \wedge a_m \in V_m)$ if and only if $o[a_i] \in V_i, \forall i \in [1, m]$.

Definition 3.1. *A* rule *r is an expression of the form $B_0 \wedge \neg B_1 \wedge \cdots \wedge \neg B_k \rightarrow c$, where $B_0, \ldots, B_k$ are conjunction of conditions and c is a class. $B_0 \wedge \neg B_1 \wedge \cdots \wedge \neg B_k$ is called* body *of the rule, whereas c is the* head *of the rule, denoted as $r.class$.*

The size of a rule $r : B_0, \wedge \neg B_1 \cdots \wedge \neg B_k \rightarrow c$, denoted as $|r|$, is the number of conditions in B_0. An object o of D is *activated* by a rule $r : B_0 \wedge \neg B_1 \wedge \cdots \wedge \neg B_k \rightarrow c$, if and only if o satisfies the positive component, B_0, and does not satisfy any negative component, $B_1, \ldots, B_k$. The set of objects of a dataset D activated by a rule r is denoted as $r(D)$. The size $r(D)$ is called *support* of the rule and denoted as $\sigma(r)$.

Given h rules, $r_1, \ldots, r_h$, they are said to be *same-head* rules if for each pair of rules r_i, r_j it holds that $r_i.class = r_j.class$.
The size of a rule $r : B_0, \wedge \neg B_1 \cdots \wedge \neg B_k \rightarrow c$, denoted as $|r|$, is the number of conditions in B_0.

A rule r is *exclusive* with respect to a a rule r' on the dataset D, if no object in D activated by r is activated also by r', namely if $r(D) \cap r'(D) = \emptyset$.

The objects activated by a rule $r : Body \rightarrow c$ whose class is actually c are called *true positive*, the other objects activated by r are called *false positive*.

Definition 3.2. *Let r : Body $\rightarrow c$ be a rule on a dataset D labeled w.r.t. a set of labels $\mathcal{C}$. The confidence of r, denoted as $\gamma(r)$ is the ratio between the true positive objects activated by r and the support of r.*

The above notions, given for a single rule, can be naturally extended to a set of same-head rules. The set $\mathcal{R}(D)$ of objects activated by a set of same-head rules $\mathcal{R}$ is the set of objects activated by at least one rule $r \in \mathcal{R}$. A rule r is exclusive with respect to a set of same-head rules $\mathcal{R}$ if and only if $r(D) \cap \mathcal{R}(D) = \emptyset$. The support of a set of same-head rules is $\sigma(\mathcal{R}) = |\mathcal{R}(D)|$, while the confidence $\gamma(\mathcal{R})$ is the ratio between the true positive objects activated by $\mathcal{R}$ and the support of $\mathcal{R}$.

Finally, the $\overline{\wedge}$ operator is introduced. Let $r : B_0 \wedge \neg B_1 \wedge \cdots \wedge \neg B_k \rightarrow c$ be a rule and $r' : B'_0 \rightarrow c$ be a rule with the same head as r and without negative components. $r \,\overline{\wedge}\, r'$ denotes the rule $r'' : B_0 \wedge \neg B_1 \wedge \cdots \wedge \neg B_k \wedge \neg B'_0 \rightarrow c$. In other words, $r \,\overline{\wedge}\, r'$ produces a rule that activates all the objects activated by r and not by r', and then r'' is exclusive with respect to r'.

The main problem we aim to solve can hence be formally defined as follows.

Definition 3.3. *Given a dataset D, a scoring function $\mathcal{F}$, three thresholds $\sigma^{\min}$, $\gamma^{\min}$ and X, the problem is:* find the set $\mathcal{R}$ of rules each having at least confidence $\gamma^{\min}$ and support $\sigma^{\min}$ such that $|\mathcal{R}(D)|$ is as close to X as possible, and the objects in $\mathcal{R}(D)$ score the highest values of $\mathcal{F}$.

In our framework, the attribute C is set to separate the highest values of $\mathcal{F}$, as described in the previous section.

Since the goal is to provide an intelligible explanation together with the list of top defrauders, the idea is to build a rule-based classifier, so that the set of rules employed for the classification directly provides a description of the fraudulent individuals.

Nevertheless, these rules could activate more than X objects in the dataset, and then only a subset of them should be employed to build the final classifier. Moreover, the global performance of the set of rules must be measured to evaluate if it is "good" enough. Such evaluation is based on the global support, namely the number of objects activated by the whole set of rules, which must be as close to X as possible, and the global confidence. Such a topic will be addressed later in this section. If the performance is not good, an other classifier is trained, and the novel rules are merged with the previous ones. Then, a new selection phase, followed by a new evaluation phase are performed. Hence, a novel classifier is trained until the global performance reaches the desideratum.

All these phases, namely the extraction of the rules, the merge of rules coming form different classifier, the selection of the best rules, and finally the evaluation of the global classifier, present some interesting aspects and they are the topics of the following paragraphs.

3.5.1 Generating rules

The first step consists in extracting the rules from the training set. The adoption of a single classifier directly trained over the training is infeasible: as shown in figure 3.6, the preprocessed training set is highly unbalanced w.r.t. the top class, and even the adoption of advanced mechanism for resampling or cost-sensitive learning produces low-accuracy models. The reason for this phenomenon is well-known in literature as the problem of *rare cases*. The latter are very small portions of the training data, that can be viewed as exceptional sub-concepts seldom occurring within predominant or rare classes. In the VAT refund fraud scenario, this corresponds to the fact that each defrauder has a peculiar behavior that does not generalize to other defrauders. As pointed out in [117], rarity actually prevents a rule-based classifier from finding and reliably generalizing the regularities within infrequent classes and exceptional cases. Indeed, due to the commonly adopted metrics for growing classification rules and evaluating their accuracy, class imbalance leads to several accurate rules targeting the predominant classes, supplemented by very few (if any) error-prone rules predicting minority classes. Furthermore, rare cases tend to materialize within the resulting classifier as strongly inaccurate rules, referred to as *small disjuncts* [64], that in most cases do not generalize actual exceptions, being a mere consequence of noise in the training data [118] . In highly imprecise learning settings, noise often contributes to the effects of rarity on predictive accuracy. On one hand, it may further skew the already unbalanced class distribution. On the other hand, rare cases may appear to the learner as indistinguishable from noise, thus requiring a more specific inductive bias, that would ultimately also induce noisy small disjuncts.

The solution provided by Sniper consists in building a hybrid classifier, resulting from the combination of the whole set of classifiers trained over the training set. The approach is similar in spirit to a bagging methodology [15]. However, rather than implementing a voting mechanism over an independent set of similar rule-based models, we chose to decompose each classifier into a single ruleset and to merge all the rulesets into a global ruleset $\mathcal{R}$, from which to extract the most prominent rules.

Decoupling the model construction phase from model selection provides us with the further advantage of approaching the *rare case* problem with a brute-force approach: in the model construction, several different strategies are attempted to build models specialized on local peculiarities of the top

class. In the model selection phase, several local fragments can be combined or discarded if the global accuracy improves.

3.5.2 Merging rulesets

Let $R_1, \ldots, R_h$ be the set of rules returned by h classifiers, and let *top* be the class label assigned by the classifiers to the objects belonging to the top class. The candidate ruleset $\mathcal{R}$ is defined as follows:

$$\mathcal{R} = \left\{ r \in \bigcup_{i \in [1,h]} R_i \mid r.class = top \right\}$$

The ruleset $\mathcal{R}$ still represents a classifier, and class *top* is assigned to a non-labeled object o if and only if there exists at least a rule in $\mathcal{R}$ that activates it. Hence, all and only the objects in $\mathcal{R}(D)$ are labeled *top*.

Taken as a whole, the global ruleset $\mathcal{R}$ presents two relevant shortcomings. first, $|\mathcal{R}(D)|$ can be larger than X. Second, the confidence of $\mathcal{R}$ can be too low, and in particular, it could be lower than $\gamma_{\min}$. Indeed, $\mathcal{R}$ is the result of merging different and independently designed classifiers which are not necessarily exclusive.

Assume, for the sake of simplicity, that $\mathcal{R}$ is composed by only two rules r_1 and r_2 having confidence $\gamma(r_1) = \frac{p_1}{p_1+n_1}$ and $\gamma(r_2) = \frac{p_2}{p_2+n_2}$, respectively. Here, p_i (resp., n_i) denotes the number of true (resp., false) positive objects activated by the rule r_i.

Let $p_{1,2}$ (resp., $n_{1,2}$) be the number of true (resp., false) positive objects activated by both r_1 and r_2; with $p_{1,2}$ ranging in $[0, \min\{p_1, p_2\}]$ and $n_{1,2}$ ranging in $[0, \min\{n_1, n_2\}]$. Then, the global confidence of $\mathcal{R} = \{r_1, r_2\}$ is:

$$\gamma(\mathcal{R}) = \frac{p_1 + p_2 - p_{1,2}}{p_1 + n_1 + p_2 + n_2 - p_{1,2} - n_{1,2}}.$$

Hence, the maximum value of $\gamma(\mathcal{R})$ is obtained when $p_{1,2} = 0$ and $n_{1,2} = \min\{n_1, n_2\}$,

$$\gamma_{\max}(\mathcal{R}) = \frac{p_1 + p_2}{p_1 + n_1 + p_2 + n_2 - \min\{n_1, n_2\}},$$

which is the case when the sets of true positive objects activated by r_1 and r_2 are disjoint, whereas the sets of false positive objects activated by r_1 and r_2 are overlapped.

Conversely, the minimum value of $\gamma(\mathcal{R})$ is obtained when $p_{1,2} = \min\{p_1, p_2\}$ and $n_{1,2} = 0$,

$$\gamma_{\min}(\mathcal{R}) = \frac{p_1 + p_2 - \min\{p_1, p_2\}}{p_1 + n_1 + p_2 + n_2 - \min\{p_1, p_2\}},$$

which is the case when the sets of true positive objects activated by r_1 and r_2 are overlapped, whereas the sets of false positive objects activated by r_1 and r_2 are disjoint.
It is worth pointing out that $\gamma_{\max}(\mathcal{R})$ can be larger than $\max\{\gamma(r_1), \gamma(r_2)\}$, whereas $\gamma_{\min}(\mathcal{R})$ can be smaller than $\min\{\gamma(r_1), \gamma(r_2)\}$. This depends from both the confidences and the supports of the rules r_1 and r_2.
In case the rules are exclusive, both $p_{1,2}$ and $n_{1,2}$ are equal to 0. Then,

$$\gamma(\mathcal{R}) = \frac{p_1 + p_2}{p_1 + n_1 + p_2 + n_2}.$$

Suppose, w.l.o.g., that $\gamma(r_1) < \gamma(r_2)$, then $\frac{p_1}{p_1+n_1} < \frac{p_2}{p_2+n_2}$. It follows that:

$$\gamma(\mathcal{R}) = \frac{p_1 + p_2}{p_1 + n_1 + p_2 + n_2} > \frac{p_1 + p_2}{p_1 + n_1 + \frac{p_2}{p_1}(p_1 + n_1)} > \frac{p_1}{p_1 + n_1}$$

Analogously, it can be shown that $\gamma(\mathcal{R}) < \frac{p_2}{p_2+n_2}$.

Summarizing, if the rules are exclusive the value of the global confidence $\gamma(\mathcal{R})$ is greater than the minimum confidence of the rules in $\mathcal{R}$ and lower than the maximum confidence of the rules in $\mathcal{R}$; conversely, these properties do not hold if the rules are not exclusive.

Thus, $\mathcal{R}$ is not necessarily the optimal choice for the final binary classifier. We can, however, look for an optimal subset $\mathcal{R}^* \subset \mathcal{R}$, which simultaneously reaches the two following goals: (i) the number of objects retrieved in the dataset is as close to X as possible, and (ii) the confidence of $\mathcal{R}^*$ is as high as possible.

The search for the best subset $\mathcal{R}^*$ achieving these two goals is referred to as SBR problem in the following. Solving such a problem is a hard task. Unfortunately, the SBR problem is NP-hard. Details of the proof can be found

Input: A set of non-exclusive positive rules $\mathcal{R}$,
a confidence threshold $\gamma_{\min}$,
an integer X
Output: A model $\mathcal{M}$
Method:
1: $\mathcal{M} := \emptyset$
2: $\mathcal{R} := \left\{ r \in \mathcal{R} \mid \gamma(r) \geq \gamma_{\min} \right\}$
3: **while** $\mathcal{R} \neq \emptyset$ **do** *//first stop condition*
4: $r^* := \arg\max_{r \in \mathcal{R}} \left\{ \gamma(r) \right\}$ *//select the best rule*
5: $\mathcal{M} := \mathcal{M} \cup \{r^*\}$ *//update the current model*
6: **if** $\mathcal{M}(D) \geq X$ **then** *//second stop condition*
7: **return** $\mathcal{M}$
//update the set of rules
8: $\mathcal{R} := \left\{ r' = r \barwedge r^* \mid (r \in \mathcal{R} \setminus \{r^*\}) \bigwedge (\gamma(r') \geq \gamma_{\min}) \right\}$
9: **return** $\mathcal{M}$

Fig. 3.7: Selecting Best Rules Algorithm

in [7].

In this section we describe a greedy technique for obtaining the resulting ruleset, starting from $\mathcal{R}$. Loosely speaking, the heuristic employed consists in iteratively taking the most confident rules until X objects are retrieved from D, or until no further rules with enough confidence exist in $\mathcal{R}$.

The algorithm is shown in figure 3.7. We employ the term *set of rules* to refer to the input set of same-head rules coming from the classifiers, whereas the term *model* refers to the set of rules finally computed by the algorithm. The main idea is to compute a model $\mathcal{M}$ by iteratively adding the most confident rule to it. Since rules may overlap, the confidence of the rules is evaluated with regards to the objects not activated by the model $\mathcal{M}$ associated with the current iteration, rather than to the whole test set.

First of all, the algorithm removes from the input set $\mathcal{R}$ those rules that are not at least $\gamma_{\min}$ confident. Then, the most confident rule r^* in $\mathcal{R}$ is selected and added to $\mathcal{M}$ (*lines* 4-5). Next, the set $\mathcal{R}$ is updated by removing r^* and by replacing each rule r other than r^* with the rule $r' = r \barwedge r^*$ if $\gamma(r') = \gamma_{\min}$, otherwise r is just removed from $\mathcal{R}$ (*line* 8).

In such a way, the rules which are now in $\mathcal{R}$ can activate only objects which are not simultaneously activated by any other rule in $\mathcal{M}$. In other words, each rule in $\mathcal{R}$ is exclusive with respect to the set $\mathcal{M}$ of rules.

The main property of the algorithm consists in the fact that, for a given rule r, the more the set of true positives activated overlaps with the true positives activated by $\mathcal{M}$, the higher the confidence of $r' = r \,\overline{\wedge}\, r^*$ is. Hence, each iteration selects the rule that gives the best contribution to the global confidence of the model $\mathcal{M}$. Moreover, since at each iteration a rule is added to $\mathcal{M}$ which is exclusive with respect to $\mathcal{M}$, and since the confidence of such a rule is at least $\gamma_{\min}$, the confidence of $\mathcal{M}$ cannot be lower than $\gamma_{\min}$. This guarantees that the heuristic produces a high-quality model.

The algorithm proceeds until one of the two stopping conditions is reached, namely either no other rule is in $\mathcal{R}$ (*line* 3), or $\mathcal{M}$ activates X objects in the dataset (*line* 6).

Formally,

$$\mathcal{R}^* = \arg\min_{S} \Big\{ |S| \;\; s.t. \;\; S \subseteq \mathcal{R} \wedge |S(D)| \geq X \wedge \\ \forall s \in S, \forall r \in \mathcal{R} \setminus S \quad \gamma(s) \geq \gamma(r) \Big\}$$

In words, $\mathcal{R}^*$ is the smallest subset of rules of $\mathcal{R}$ able to activate at least X objects of the dataset and such that there is not a rule in $\mathcal{R} \setminus \mathcal{R}^*$ more confident than any rule in $\mathcal{R}^*$.

As far as the evaluation of the global classifier is concerned, the confidence of the whole set of rules is computed by means of the well-known *cross validation*, hence by exploiting the training set.
Then, the approach here proposed employs the confidence threshold $\gamma^{\min}$ to evaluate the accuracy of the current model. If its confidence does not exceed the threshold, it is enhanced by training a novel classifier.
The appendix C shows a simple example of how the algorithm performs the merging of the rules.

3.6 Results

In this section, we briefly show the main experimental results obtained applying the Sniper technique to the real-life VAT refund fraud scenario so far considered. The results are still preliminary, as the technique is being validated on real-life scenarios.

3.6.1 Learning of single classifiers

First of all, we have separately computed several classification models using a score function to label the examples belonging to training set. Precisely, we have preferred to exploit the score function based on the $\mathcal{F}_{\Pi}$ function in order to better fit the domain's constraints. The classifiers have been selected from the Weka workbench [120, 123] and other commercial tools. Several different parameters sets were adopted, including cost models for cost-sensitive learning. The classifiers have been selected from the Weka workbench obtained by the execution of algorithms *C4.5* [94], *Ripper* [31] and *PART* [123], employing *WEKA* [123] and *Clementine* [104] as implementation tools. The results of some experiments are reported in the first part of table 3.1. The experiments marked with a "$*$" refer to classifiers modified in order to improve their performance in terms of subjects retrieved. That is, if the underlying original classifier extracts more than X subjects, the less confident rules are removed until a number of subjects close to X is retrieved from the dataset. Note that since all the algorithms employed extract a model with exclusive rules, if the less confident rule is removed from the model, the global confidence rises.

For each classifier C_i, the table contains information about the support and the confidence of the model extracted by C_i on the test set (*columns 2-3*); and finally, the number of subjects of the dataset identified as fraudulent by C_i. The classifiers are ordered by increasing value of confidence.

None of the single classifiers satisfies our quality needs. Indeed, they are not able to simultaneously ensure a small number of false positives and a number of dataset subjects retrieved close to $X = 10{,}000$. In particular, high-quality models are only capable of selecting a small number of subjects from the whole population, which is too far from the value X required. Conversely, larger auditing sets can only be obtained by low-accuracy classifiers.

3.6.2 Sniper technique results

Then, we have ran the Sniper technique considering as input the set of rules containing all the rules of each classifiers C_i above reported. The parameters we adopt in the experiments are $\sigma_{\min} = 0.1\%$ (corresponding to 50 subjects), $\gamma_{\min} = 70\%$, and $X = 10{,}000$.

classifier	*supp (%)*	*conf (%)*	*dataset subjects*
C_1	1.01	84.90	1,910
C_2	1.10	82.97	2,240
C_3	3.11	77.28	4,955
C_4	3.44	77.12	5,675
C_5^*	6.36	62.26	10,056
C_6^*	6.81	60.80	8,875
C_7^*	7.07	59.72	9,059
C_8^*	5.22	52.64	9,950
C_9^*	4.56	49.18	12,584
S	8.78	80.41	9,840

Table 3.1: Single classifiers vs Sniper classifier

rule	*1*	*2*	*3*	*4*	*5*	*6*	*7*
supp	0.65	1.21	0.97	0.89	0.85	0.87	0.90
conf	97.64	94.53	88.41	88.09	87.76	87.66	87.29
rule	*8*	*9*	*10*	*11*	*12*	*13*	*14*
supp	1.01	0.12	0.17	0.52	0.26	0.17	0.19
conf	85.12	83.64	83.12	77.73	76.47	71.79	70.11

Table 3.2: Rules of the final classifier

Notice that, according to described approach, the mechanism governing rules' selection guarantees that the returned set of rules is characterized by a global confidence value greater than the fixed threshold $\gamma_{\min} = 70\%$.

The final result of the procedure is the model denoted as S in the second part of table 3.1. Precisely, S contains 14 rules coming from 9 distinct single classifiers. Table 3.2 shows the characteristics of each of these rules.

The global confidence of the final model $\mathcal{S}$ is $\gamma(\mathcal{R}) = 80.41\%$; whereas the number of total subjects of the dataset it activates is 9,840. Such results give

support that the Sniper technique outperforms single classifiers with respect to the considered constraints.

4

Exceptional Fraudsters Detection

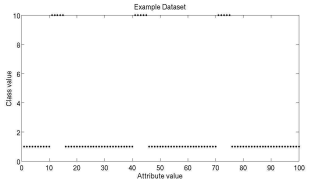
4.1 Problem definition

In real-world scenarios it is not always possible to identify a single threshold to separate interesting fraudsters from the others. Nevertheless in such cases are required the domain expert skills to validate the choice threshold. In this chapter an approach, called Numeric SNIPER , is introduced for learning an intelligible and compact model of object exceptionality, that is capable to effectively and efficiently deal with a continuous target attribute. However, differently from regression problems, given an input dataset (i.e. a collection of object records) over some suitable attribute schema, the task is not to simply predict the value of the target attribute for the individual objects in the dataset, but to identify a subset of these objects that mostly show an exceptional behavior. In this regard, the target attribute can be explained as some domain-specific measure, that associates each object in the dataset with an extent of behavior exceptionality. The values of the target attribute for the objects in the dataset are unknown. Nevertheless, it is assumed the availability of a separate set of objects (over the same attribute schema), called *training set*, for which the values of the target attribute are known. The exceptional behavior is measured by means of a function that associates a value representing the exceptionality with an object. Clear enough, this function is not known for the objects of the dataset. Nevertheless, it is assumed that a set of objects, called *training set*, for which such a function is known is available. The idea behind Numeric SNIPER is essentially to learn from the training data a set of rules through a sequential covering scheme. The resulting rule set is then applied to the dataset at hand in order to retrieve a required number of objects, which are "mostly similar" to the more exceptional objects within the training set, namely which are more likely to score the maximum value of exceptionality in the dataset. The Numeric SNIPER learning process repeatedly learns one rule from the training objects, adds it to a rule set and excludes from further consideration the covered training objects. This process halts when a suitable compromise between the complexity and the fit of the

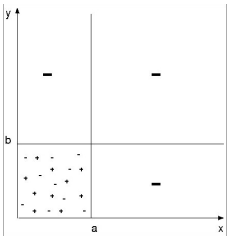
rule-set is reached. The individual rules are learnt as conjunctions of suitable conditions on the attributes of the training objects. In particular, each rule is learnt through a greedy scheme, that progressively adds individual conditions that guarantee the maximum increase in the accuracy of the rule. The latter is evaluated through a novel method, that essentially accounts for the deviation of the target attribute in the covered training objects from the maximum value of exceptionality in the training set. In principle, three categories of approaches from the current literature might be chosen for application in the envisaged scenario, i.e. *regression*-based, *model trees*-based and *classification*-based methods.

In particular, prediction approaches based on the statistical techniques of regression [61] aim to predict the value of a continuous attribute, say $\mathcal{F}$, by means of a linear or nonlinear mathematical model that involves one or more predictor attributes in the schema of the training data. Depending on the specific applicative setting and requirements, two major forms of regressions can be used for prediction, i.e. linear and nonlinear. Linear regression is the simplest form, that models the class attribute $\mathcal{F}$ by means of a linear model, i.e. as some linear function of one or more attributes in the schema of the training data. Precisely, in bivariate regression, the linear model is a straight-line involving one predictor attribute. Model parameters (i.e. line slope and $\mathcal{F}$ intercept) are usually estimated via the so-called method of least squares. In multiple regression, the linear model is a refinement of the straight-line model that involves multiple predictor attributes. Nonlinear regression extends the basic linear model by adding polynomial terms. There are several important differences between regression and the envisaged task, both in terms of quality of the results and in terms of computational efficiency. As far as the quality of results is concerned, regression-based methods are meant for building a global model, which minimizes the prediction error for each object in the dataset. This represents a major limitation for their application to our purpose, since often the values of the target $\mathcal{F}$ attribute can be very hard to fit in a global function. Instead, the individual rules of the model yielded by Numeric SNIPER are capable to identify (possibly small) sets of objects, which are likely to score high values of $\mathcal{F}$. Thus, Numeric SNIPER performs much better in those scenarios where the objects to be identified are rare and dispersed, or their behaviors do not fit in a common pattern. The major limitation behind the exploitation of regressors in the above delineated setting is that these are meant for building a global model, which minimizes the prediction error for each object in the dataset. Nevertheless, in some cases $\mathcal{F}$ can be very hard to fit in a global function. These observations are elucidated in fig. 4.1a. The example dataset consists of a single attribute a, and 100 objects. The value that the objects assume on a is reported on the horizontal axis, whereas the value of the continuous class associated with the objects is reported on the vertical axis. The rule set $\mathcal{R} = \{r_1, r_2, r_3\}$, where $r_1 : 10 \leq x \leq 15$, $r_2 : 40 \leq x \leq 45$ and $r_3 : 70 \leq x \leq 75$, is an example of a simple and optimum model for

identifying the objects scoring the highest values of $\mathcal{F}$ within the illustrated scenario. Conversely, training a regressor to model the trend of the data is very challenging. Regressors are also inadequate for the envisaged task from a computational-efficiency point of view, since merely predicting the value of $\mathcal{F}$ for each object in the dataset does not suffice to single out the objects scoring the highest values of $\mathcal{F}$. A further step for ordering the objects and selecting them by their predicted $\mathcal{F}$ values is required.



(a)



(b)

$\mathbf{a}_1$	$\mathbf{a}_2$	$\mathbf{a}_3$	$\mathcal{F}$	C
a	b	d	10	1
a	b	d	10	1
a	b	e	8	0
a	b	e	8	0
a	b	e	8	0
a	c	d	1	0
a	c	d	1	0

(c)

Fig. 4.1: Example settings revealing the inadequacy of regressors (4.1a), model trees (4.1b) and classification methods (4.1c).

Model trees [63,89,93,113] also aim to predict the value of function $\mathcal{F}$ and not to identify the objects maximizing $\mathcal{F}$. From this perspective, the goal of such methods is again to minimize the prediction error, which is relevantly different from ours. Essentially, model trees attempt at partitioning the feature space into regions, such that the objects within the same region well fit in some regression model. This is problematic in our context whenever a region, such as the one in fig. 4.1b, includes objects that behave differently according to $\mathcal{F}$ and some dispersed objects in the whole region score a (nearly) maximum value of $\mathcal{F}$. Indeed, while in such cases a simple rule (such as $x \leq a \wedge y \leq b$ for the illustrated setting) is capable at identifying the objects scoring the maximum values of $\mathcal{F}$, a model tree would either miss the whole region, or predict far too low values for the objects falling in the region. In neither case would the objects likely to score a (nearly) maximum value of $\mathcal{F}$ be identified.

4.2 Notation

In this section, we provide a proper notation, some preliminary concepts and notions, that represent the basics of our approach.

An *attribute* a is a descriptive feature with an associated domain denoted as $Dom(a)$. Let $\mathcal{A} = \{a_1, \ldots, a_m, \mathcal{F}\}$ be a schema composed of a set of $m+1$ attributes, wherein $\mathcal{F}$ is real-valued attribute (i.e. $Dom(\mathcal{F}) = \Re$), whose value provide an extent of the exceptionality of the objects defined over $\mathcal{A}$. $Dom(\mathcal{A})$ is the domain $Dom(a_1) \times \cdots \times Dom(a_h) \times \Re$. A training set $\mathcal{T} = \{\mathbf{o}_1, \ldots \mathbf{o}_n\}$ is some collection of *objects* from $Dom(\mathcal{A})$. More precisely, the generic object $\mathbf{o} \in \mathrm{d}$ is the $m+1$-ple $\mathbf{o} = \langle v_1, \ldots, v_m, \mathcal{F}(\mathbf{o})\rangle$ of m values, where $v_i \in Dom(a_i)$ and $\mathcal{F}(\mathbf{o})$ indicates the value of the $\mathcal{F}$ attribute in the context of the object $\mathbf{o}$.

A *condition* c on $\mathcal{A}$ is an expression of the form $a \in V$, where $a \in \mathcal{A}$ and $V \subseteq Dom(a)$. In particular, if a is a categorical attribute, V becomes a singleton value from $Dom(a)$. Otherwise, if a is a numerical attribute, V is some suitable interval. In such a case, the condition $V = Dom(a)$ is called *trivial*, since it is always satisfied.

The total number of conditions (including the trivial one) that can be expressed for an attribute a in $\mathcal{A}$ is determined as follows. Let n_a be the number of different values that a assumes on $\mathcal{T}$ (then, $n_a \leq |\mathcal{T}|$). If a is a categorical attribute then $n_c(a) = n_a + 1$. Conversely, if a is a numerical attribute, then the total number of conditions which can be defined on a is given by taking into account (i) the total number of bounded intervals, $\frac{(n_a)\cdot(n_a+1)}{2}$, (ii) the total number of left-unbounded and right-unbounded intervals, $2 \cdot n_a$, (iii) the trivial condition. Therefore, $n_c(a) = \frac{(n_a)\cdot(n_a+1)}{2} + 2n_a + 1 = \frac{(n_a+2)\cdot(n_a+3)}{2} - 2$. Given a training set $\mathcal{T}$ over a schema $\mathcal{A}$, the space of all possible conditions on the individual attributes of $\mathcal{A}$ is denoted as $C_{\mathcal{T}}$.

A *rule* over $\mathcal{T}$ is a conjunction of non-trivial conditions from $C_{\mathcal{T}}$. The size of a rule r, denoted as $|r|$, is the number of conditions in r. An object $\mathbf{o}$ of $\mathcal{T}$ is *activated* by a rule $r : (a_1 \in V_1) \wedge \cdots \wedge (a_h \in V_h)$ if and only if $\mathbf{o}[a_i] \in V_i, \forall i \in [1, h]$. The subset of objects in $\mathcal{T}$ activated by a rule r is denoted as $r(\mathcal{T})$. The size of $r(\mathcal{T})$ is the *support* of rule r. Furthermore, an object $\mathbf{o} \in \mathcal{T}$ is said to be activated by a set of rules $\mathcal{R}$ if and only if it is activated by at least one rule $r \in \mathcal{R}$. More precisely, the set of objects activated by a set of rules $\mathcal{R}$ is $\mathcal{R}(\mathcal{T}) = \bigcup_{r\in\mathcal{R}} r(\mathrm{d})$. The size of $\mathcal{R}(\mathcal{T})$ is the *support* of the rule set $\mathcal{R}$.

Numeric SNIPER learns a model of object exceptionality, i.e. a disjunction $\mathcal{R}$ of conjunctive rules, from a training set T. The resulting rule set $\mathcal{R}$ can then be applied to any dataset d (i.e. to any collection of objects for which the values of attribute $\mathcal{F}$ are unknown) for the purpose of identifying those objects that are likely to score the maximum values of $\mathcal{F}$ in d.

The notion of rule accuracy is central to the induction of model $\mathcal{R}$ of object exceptionality. It is proposed a novel notion of accuracy for a rule, that is related to the value that the $\mathcal{F}$ attribute assumes over the individual objects activated by the rule. Intuitively, the devised notion of rule accuracy is based on the *weight* of the objects in the training set, according to the $\mathcal{F}$ attribute. In turn, the weight of the generic object is proportional to the value that the same object assumes on $\mathcal{F}$: the higher the weights of the objects activated by a rule r, the higher its accuracy.

More specifically, let $\mathcal{T}$ be a training set and $\mathcal{F}$ the measure of exceptionality observed in $\mathcal{T}$.

Definition 4.1. *The* weight *of an object* $\boldsymbol{o} \in \mathcal{T}$ *with respect to* $\mathcal{F}$ *is*

$$w(\boldsymbol{o}) = 1 - \frac{\mathcal{F}^{\max} - \mathcal{F}(\boldsymbol{o})}{\mathcal{F}^{\max} - \mathcal{F}^{\min}}$$

where $\mathcal{F}^{\max}$ (resp. $\mathcal{F}^{\min}$) denotes the maximum (resp. the minimum) value of $\mathcal{F}$ in $\mathcal{T}$. By definition, the weight of an object $\mathbf{o}$ always ranges between 0 and 1: the closer the value of $\mathcal{F}$ scored by $\mathbf{o}$ to $\mathcal{F}^{\max}$ in $\mathcal{T}$, the higher the weight of $\mathbf{o}$. The weights of the objects activated by a rule r are, then, employed to measure the *strength* $\theta(r)$ of r as reported below

$$\theta(r) = \sum_{\mathbf{o} \in r(\mathcal{T})} w(\mathbf{o})$$

The strength of a rule r measures the effectiveness of r in activating objects scoring high values of $\mathcal{F}$. By building on strength, the accuracy of a rule can be defined as follows.

Definition 4.2. *Let* r *a rule over training set* $\mathcal{T}$*. The* accuracy $\gamma(r)$ *of* r *is*

$$\gamma(r) = \frac{\theta(r)}{|r(\mathcal{T})|}$$

By definition 4.2, $\gamma(r)$ is high when the strength of r is high, namely when r activates objects scoring high values of $\mathcal{F}$. Nevertheless, the larger the number of objects activated by r, the smaller the value of $\gamma(r)$. In particular, $\gamma(r)$ is 1 if and only if r activates only those objects scoring the maximum value of $\mathcal{F}$. In principle, this might be susceptible to overfitting. The latter issue is effectively dealt with in subsection section 4.3.2, by means of a robust mechanism adopted in the learning process to trade off the compactness of the learnt set of rules with its fit.

To conclude, we highlight that, in principle, a connection can be established between the devised notion of rule accuracy and the traditional notion of rule confidence. To elucidate, assume that a training set $\mathcal{T}$ is partitioned in two classes, respectively called 0 and 1. Let $\mathcal{F}$ value 0 for the objects belonging to class 0 and 1 otherwise. In addition, let r be some conventional rule targeting class 1, that activates m objects of class 1 and $n-m$ objects of class 0. Then, the confidence of r is $\frac{m}{n}$. Furthermore, since the weight of each object within class 1 amounts to 1, whereas 0 is the weight of the objects within class 0, the accuracy of r according to definition 4.2 is $\frac{m}{n}$, which is the same as the confidence of r. This intuitively provides an interpretation of accuracy as the generalization of confidence to a continuous target attribute.

4.3 Numeric SNIPER Technique

In this section we discuss the Numeric SNIPER technique. As already anticipated, the aim is to identify a subset of objects in a dataset d, that mostly score an exceptional behavior according to the (unknown) values of a continuous attribute $\mathcal{F}$ (modeling some measure of object exceptionality) in the schema of d. The approach consists in learning a set of rules from a training set $\mathcal{T}$ (whose objects are associated with known values of the foresaid $\mathcal{F}$ attribute), that can then used to identify the objects that are likely to score the maximum values of $\mathcal{F}$ in d. The resulting rule-set should be optimum with respect to a suitable complexity criterion. The basic concepts used to formulate a notion of model optimality in terms of both rule-cost and model-cost are provided in subsection section 4.3.1. The Numeric SNIPER algorithmic scheme is then covered in subsection section 4.3.2.

4.3.1 Rule and Model Cost

Numeric SNIPER builds an ordered rule set $\mathcal{R}$ by repeatedly appending to $\mathcal{R}$ one rule r at a time. The individual rule r is appended only if it does not increases model complexity up to the point of potentially overfitting the underlying training data. The Minimum-Description-Length principle [57, 87] is used to establish a suitable compromise between model compactness and fit. The overall description length for the resulting rule set relies on the cost (i.e. the description length) of the individual rules. We next provide a definition for the cost of one rule and then extend such a definition to a set of rules. Given a training set $\mathcal{T}$ over an attribute schema $\mathcal{A}$ and a rule r on $\mathcal{T}$, the cost of r is measured in terms of the total length of a rule, which takes into account two components: (i) the length of the encoding of the rule, and (ii) the length of the encoding of the data given the rule. The first component can be obtained by means of the following formula:

$$length(r) = \log_2 \left(n_c^{|r|} \binom{|\mathcal{A}|}{|r|} \right) \tag{4.1}$$

The above equation assumes that a same number n_c of conditions can be defined for each attribute $a \in \mathcal{A}$. Recall that n_c is $\frac{(|\mathcal{T}|+2)\cdot(|\mathcal{T}|+3)}{2} - 2$ for continuous attributes as defined in section 4.2. Indeed, the first condition can be chosen among $n_c \cdot |\mathcal{A}|$ conditions, the second condition can be chosen among $n_c \cdot (|\mathcal{A}| - 1)$ conditions, ..., the $|r|$-th condition can be chosen among $n_c \cdot (|\mathcal{A}| - |r| + 1)$ conditions and so forth. Overall, there are $(n_c \cdot |\mathcal{A}|) \cdot (n_c \cdot (|\mathcal{A}| - 1)) \cdot \cdots \cdot (n_c \cdot (|\mathcal{A}| - |r| + 1))$ rules having size $|r|$, which can be more succinctly rewritten as $n_c^{|r|} \frac{|\mathcal{A}|!}{(|\mathcal{A}|-|r|)!}$. However, the latter result accounts for multiple orderings of the same set of conditions across the rules. To obtain the exact number of rules with size $|z|$, without accounting for all possible orderings of the conditions, a division by $|r|!$ is required. Hence, we obtain that the total number of rules with size $|r|$ is $n_c^{|r|} \cdot \binom{|\mathcal{A}|}{|r|}$. The encoding of one such a rule requires $log_2 \left(n_c^{|r|} \cdot \binom{|\mathcal{A}|}{|r|} \right)$ bits.

The second component is measured by borrowing and suitably adapting the notion of *entropy* from the information theory. In particular, due to the absence of actual classes in the addressed setting, the entropy $\eta(r)$ of a rule r can be suitably redefined as follows:

$$\eta(r) = -(\gamma(r) \cdot \log(\gamma(r)) + (1 - \gamma(r)) \cdot \log(1 - \gamma(r)))$$

Essentially, $\eta(r)$ is a measure of the purity of region $r(\mathcal{T})$ (i.e. the subset of objects from $\mathcal{T}$ activated by r), in which the usual discrete classes are replaced with the extent of exceptionality for the objects in the region (that in principle form two artificial classes, namely *exceptional* and *unexceptional*).

By building on entropy, the length of the encoding of the data given rule r can be defined as the product of the entropy of r, i.e. $\eta(r)$, and its support, namely $\eta(r) \cdot |r(\mathcal{T})|$. Thus, the total cost associated with a rule r is:

$$cost(r) = length(r) + \eta(r) \cdot |r(\mathcal{T})|$$

To this point, the cost of a rule set $\mathcal{R}$ can be obtained from the cost of an individual rules as shown below:

$$cost(\mathcal{R}) = (|\mathcal{T}| - |\mathcal{R}(\mathcal{T})|) \cdot \eta_{na} + \sum_{r_i \in \mathcal{R}} cost(r_i) \tag{4.2}$$

where $(|\mathcal{T}| - |\mathcal{R}(\mathcal{T})|)$ is the number of objects not activated by $\mathcal{R}$ and η_{na} is the entropy associated with such a set.

4.3.2 Rule Learning

The Numeric SNIPER technique, sketched in algorithm 1, is a rule-learning algorithm that builds an ordered set $\mathcal{R} = \{r_1, \ldots, r_k\}$ of rules, such that the objects activated by r_i are likely to score higher value of $\mathcal{F}$ than the objects activated by r_j, if $i < j$. The rule set is learnt through sequential covering [31,103]. It is worth clarifying that when the first rule is grown, $cost(\mathcal{R}^{(1)})$ is evaluated against $cost(\mathcal{R}^{(0)})$, which is the cost of the empty rule set (defined at line 3 and simply obtainable from definition 4.2 by setting $|\mathcal{T}|$ to 0).

As far as growing the individual rules is concerned, Numeric SNIPER repeatedly adds conditions to such rules to the purpose of activating those objects (that are left) in $\mathcal{T}$, scoring the highest values of $\mathcal{F}$. This is achieved by means of the BUILD-RULE procedure. The latter initially sets each new rule r to the empty rule (at line 4) and then reiterates (at lines 8-17) the addition of conditions to r involving distinct attributes in the schema of the training data. At each step t, one condition c^* is greedily chosen (at line 12) from a set of possible conditions C (computed at line 9) as the condition that maximizes the gain for the longer and more specialized rule $r_{(t)}$ relative to the preceding rule $r^{(t-1)}$. The adopted notion of gain is suitably adapted from the information-gain measure in [31, 92]) and is represented by the *gain* function, which is defined as follows

$$gain(r^{(t)}, r^{(t-1)}) = \alpha^{(t)} \left(log(\gamma(r^{(t)})) - log(\gamma(r^{(t-1)})) \right)$$

where $\alpha^{(t)} = \sum_{\mathbf{o} \in r^{(t)}(\mathcal{T})} w^2(\mathbf{o})$. The *gain* function differs from the traditional information gain [31,92]) in one fundamental respect, i.e. no distinction is made between objects within the training data $\mathcal{T}$ with a positive or a negative class label. Rather, *gain* guides the search for a specialization of a rule $r^{(t-1)}$ into a longer $r^{(t)}$ through the addition of some condition c^* that yields an increase of accuracy in the set $r^{(t)}(\mathcal{T})$ of activated objects. In this regard, $\alpha_{(t)}$ further enforces the bias towards objects scoring high values of the $\mathcal{F}$ objective function. Notice that, once chosen, conditions are removed from $C_{\mathcal{T}}$. Therefore, when the latter becomes empty (at line 14), the rule under specialization is marked as not further improvable (at line 15). Procedure BUILD-RULE halts (at line 17) in one of two cases: either whenever the currently available rule is no more improvable, or if the greedy specialization of the rule itself does not further increase its accuracy, which occurs when $\gamma(r^{(t)}) \leq \gamma(r^{(t-1)})$. Again, it is worth clarifying that the accuracy of

each one-condition rule (i.e. $\gamma(r^{(1)})$) is evaluated against the accuracy of the empty rule, which is a sort of default rule activating all objects in $\mathcal{T}$ and whose accuracy ((i.e. $\gamma(r^{(0)})$) at line 7) is simply the average of their corresponding weights. In addition, we emphasize that computing the accuracy $\gamma(r^{(t)})$ of a rule r involves identifying the values of $\mathcal{F}^{\max}$ and $\mathcal{F}^{\min}$ scored by the objects that are left in $\mathcal{T}$ at step t. In turn, this requires to preliminarily recompute (at lines 1-3) the weights of such objects.

Algorithm 1 $BUILD - RULES(T)$

Require: A training dataset $\mathcal{T}$;
Ensure: An ordered rule-set $\mathcal{R} = \{r_1, \ldots, r_k\}$;
1: let $t \leftarrow 0$;
2: let $\mathcal{R}^{(0)} \leftarrow \emptyset$;
3: let $cost(\mathcal{R}^{(0)}) = |\mathcal{T}| \cdot \eta$;
4: **repeat**
5: $t \leftarrow t + 1$;
6: $r \leftarrow BUILD - RULE(\mathcal{T})$;
7: $\mathcal{R}^{(t)} \leftarrow \mathcal{R}^{(t-1)} \cup \{r\}$;
8: $\mathcal{T} \leftarrow \mathcal{T} - r(\mathcal{T})$;
9: **until** $cost(\mathcal{R}^{(t)}) \geq cost(\mathcal{R}^{(t-1)})$
10: **return** $\mathcal{R}^{(t-1)}$;

4.4 Evaluation

We here empirically investigate the performance of the Numeric SNIPER technique by reporting on the results of some preliminary experiments executed. We conducted tests over a selection of both real and synthetic datasets, namely `California housing` and `Bank`, that have been used extensively throughout the literature to benchmark algorithms. In particular, `California housing` is obtained from the StatLib repository and collects information on the variables using all the block groups in California from the 1990 Census. A block group on average includes $1,425.5$ individuals living in a geographically compact area. Overall, this dataset contains $20,640$ observations on 9 variables. Bank is a synthetically generated dataset (available at `http://www.cs.toronto.edu/`) of $8,192$ cases generated from a simulation of how bank-customers choose their preferred banks. It is useful for predicting the fraction of bank customers who leave the bank because of full queues. Additionally, we tested Numeric SNIPER over a further non-publicly available real-world dataset, referred to as `Fraud`, which contains $45,000$ records concerning personal demographic and fiscal information. It is used in the task of learning a model that predicts the amount of taxes presumably evaded by an individual. The main goal of our experimentation is to understand the degree of *proficiency* attained by Numeric SNIPER

Algorithm 2 $BUILD - RULE(T)$

Require: A training dataset $\mathcal{T}$;
Ensure: A single rule;
1: **for** each $\mathbf{o} \in \mathcal{T}$ **do**
2: recompute $w(\mathbf{o})$;
3: **end for**
4: let $t \leftarrow 0$;
5: let *improvable* $\leftarrow$ *true*;
6: let $r^{(0)} \leftarrow \emptyset$;
7: let $\gamma(r^{(0)}) = \frac{1}{|\mathcal{T}|} \sum_{\mathbf{o} \in \mathcal{T}} w(\mathbf{o})$;
8: **repeat**
9: let $C \leftarrow \left\{ c \in C_{\mathcal{T}} | \left(gain(r^{(t)} \cup \{c\}, r^{(t)}) > 0 \right) \right\}$;
10: $t \leftarrow t + 1$;
11: **if** $|C| > 0$ **then**
12: let $c^* \leftarrow argmax_{c \in C} \left(gain(r^{(t-1)} \cup \{c\}, r^{(t-1)}) \right)$;
13: $r^{(t)} \leftarrow r^{(t-1)} \cup \{c^*\}$;
14: **else**
15: *improvable* $\leftarrow$ *false*;
16: **end if**
17: **until** $(\gamma(r^{(t)}) \leq \gamma(r^{(t-1)}))$ || $(improvable == false)$
18: **return** $r^{(t-1)}$;

in retrieving exceptional objects. In practice, the identification of exceptional objects should not simply separate exceptional from non-exceptional objects, but it should also consider as more relevant those objects scoring higher values of exceptionality. The foresaid datasets were used to evaluate Numeric SNIPER against one representative of a consolidated category of alternative approaches, namely the m5 method [93]. The latter learns a tree-based predictive model, whose leaves can have multivariate linear models of the target attribute that essentially exploit local linearity in the training data. Both Numeric SNIPER and m5 were trained over the training data provided with the chosen datasets. The experimental methodology is as follows. Within any chosen dataset, any object activated by a specific rule of the resulting Numeric SNIPER model is marked with the identifier of the activating rule. Once marked, these objects are then sorted by the increasing values of the associated rule-identifier. A varying number of objects is then extracted from the ordered list and plotted on a graph. A point on the horizontal axis of such a graph indicates the number of exceptional objects considered. Instead, a corresponding point on the vertical axis represents the attained quality, being the sum of the extent of exceptionality over all considered objects.

It is worth noticing that, according to the adopted evaluation methodology, all objects activated by a same rule are considered equally exceptional, since no prediction is made on the actual extent of exceptionality of such objects.

Therefore, whenever considering less objects than the ones activated by a certain rule, such objects are randomly chosen from the whole set of objects activated by the rule.

A similar experimental methodology is adopted with m5. The only difference with respect to the evaluation of Numeric SNIPER is that the individual objects in the dataset are marked with (and, hence, ordered by) the extent of exceptionality predicted by the tree-based model.

In the following, we discuss on the performances achieved by Numeric SNIPER over the described datasets and compare them with those of m5.

For each dataset figures on the right report the distributions of the objective function. In particular, for a given range of the objective function $\mathcal{F}$ the correspondant bar of the histogram represents the number of objects scoring a value of $\mathcal{F}$ belonging to that range. Moreover, the aforementioned histograms give information about how the exceptional objects are distribuited with respect to the population which they belong to. For example, in the `Fraud` dataset, most of the objects score low values of $\mathcal{F}$ and very few objects score high values. On the contrary in the `California housing` dataset, the distribution of the values of $\mathcal{F}$ is more balanced.

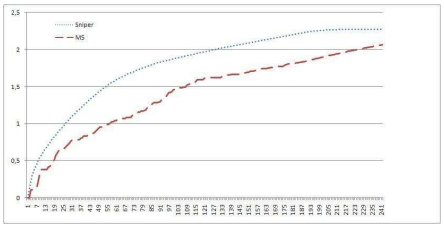
Figures highlight that Numeric SNIPER is competitive in terms of quality results with respect to m5. In particular, our approach outperforms the competitor on `Fraud` and `Bank` (Figures 4.2a and 4.2c), while on `California housing` dataset the performance of the two approaches under comparison are, in practice, indistinguishable (Figure 4.2e). This can be justified by looking at the distribution of the objective function. Indeed, in the first two datasets such a distribution is highly unbalanced (see Figures 4.2b and 4.2d), and then the exceptional objects constitute a very rare class. Conversely, in `California housing`, the objective function distribution is not clearly skewed (Figure 4.2f). As a result, the model tree approach is capable of exhibiting good performances as well, thus resulting in a similar performance with respect to Numeric SNIPER .

Therefore, on the basis of this preliminary experimental evaluation, our approach shows better performances than existing ones when the exceptional objects represents a rare class, while its performance are competitive with the other approaches in the other cases.

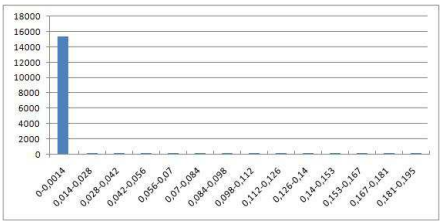
Finally, table 4.1 summarizes the computational time required by the two approaches to build the model over the considered datasets. Results clearly state that Numeric SNIPER is much more efficient than m5. In particular, our approach is one or two orders of magnitude faster than the competitor in each considered dataset.

Dataset	**Sniper**	**m5**
Fraud	21.28	1989.72
Bank	4.46	23.75
California	7.59	351.81

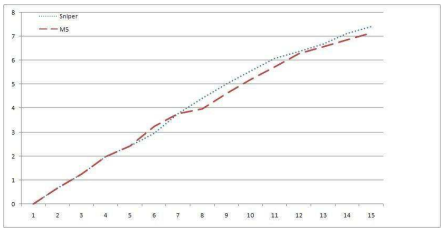
Table 4.1: Model building time in sec.



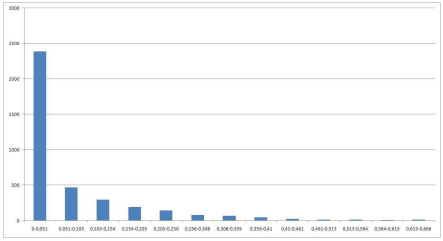
(a) `Fraud` proficiency



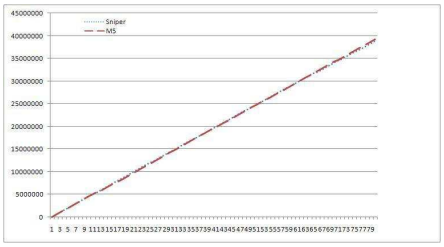
(b) `Fraud` objective function distribution



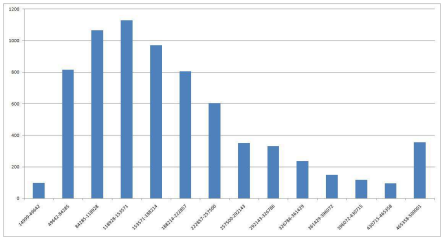
(c) `Bank` proficiency



(d) `Bank` objective function distribution



(e) `California housing` proficiency



(f) `California housing` objective function distribution

Fig. 4.2: Experimental results

5

Improving accuracy in imprecise environments

5.1 Rule-Learning with Probabilistic Smoothing

In the previous chapters it was explained that the main critical points in a data mining analisys in contexts like fraud enviroments are noise, low class separability and rarity for the classes of primary interest for the learning task. These problems are commons in other application domains. Examples, in this respect, are intrusion detection, manufacturing line monitoring [96], risk management, telecommunications management [44], medical diagnosis [24], text classification [116] and oil-spill detection in satellite images [68]. Most classification models exhibits a poor classification performance in imprecise (multi-class) learning environments. To improve the predictive accuracy, it was formalized a hierarchical classification framework for discriminating rare classes in imprecise domains. The devised framework couples the rules of a rule-based classifier with as many local probabilistic generative models. These are trained over the coverage of the corresponding rules to better catch those globally rare cases/classes that become less rare in the coverage. Two novel schemes for tightly integrating rule-based and probabilistic classification are introduced, that classify unlabeled cases by considering multiple classifier rules as well as their local probabilistic counterparts. An intensive evaluation shows that the proposed framework is competitive and often superior in accuracy w.r.t. established competitors, while overcoming them in dealing with rare classes.

5.2 Motivation

Rule learning is a mainstay of research in the field of concept learning, because of various desirable properties such as, e.g., its high expressiveness and immediate intelligibility to humans. In particular, associative classification [108] is an advance in rule learning, that relies upon the associations in the available

data between the co-occurrence of certain combinations of attribute values and their observed class labels. The resulting classification models, referred to as associative classifiers, consist of association rules, whose consequents are restricted to predict the values of the target class attribute. Associative classification retains the advantages of conventional rule learning and also tends to achieve a higher predictive accuracy [125]. Indeed, rule induction does not operate on the whole training data. Rather, it is generally performed as a heuristic separate-and-conquer process, that progressively excludes subsets of the training data from further consideration as soon as covered by locally-optimal, biased rules. Instead, associative classification yields rules with an appropriate degree of generality/specificity, that summarize the co-occurrence patterns across the whole training data.

Several approaches to associative classification are available from the literature, with differences in three major aspects, i.e discovery of classification rules, the extraction of a compact classifier and the classification of unlabeled cases [108]. Mining class association rules is a critical aspect due to the implied amount of computation. Classification rules are mined through search strategies based on Apriori [1] in [3,73,74], whereas a variant of the FP-growth algorithm [56] is used in [71]. Often, the huge number of resulting classification rules, that may overfit the training data, is pruned to distil a compact associative classifier. A variety of methods is used for this purpose, such as χ^2 testing [71], minimum class support [73], complement class support [74] as well as database coverage [3, 73, 74]. As to the classification of an unlabeled case, some methods exploit the top-quality rule covering the case [73, 74]. Other approaches take into account multiple rules applicable to the case [71, 125] and resort to suitable scoring mechanisms as well as voting.

Unfortunately, like most classification models, associative classifiers exhibit a poor predictive accuracy in highly imprecise learning settings. Additionally, cases of distinct classes may be hardly separable, which conceptually calls for classification rules with possibly (very) limited coverage and still high predictive accuracy, especially on the minority classes.

As it is pointed out in [117], rare classes originate several accurate rules targeting the predominant classes, supplemented by very few (if any) error-prone rules predicting minority classes, which are of primary interest in practical applications [119]. Rare cases, instead, tend to materialize within the resulting classifier as strongly inaccurate rules, referred to as *small disjuncts* [64]. In general, induction from rare data is likely to produce either overly specific rules, that overfit the data, or overly general rules, which do not catch the actual generalization of the covered data. Therein, learning classification rules with possibly very limited coverage and still high predictive accuracy generally involves a non trivial compromise between two contrasting aspects of rule induction, namely an appropriate degree of generalization and overfitting.

The poor performance of the classifiers produced by the standard machine learning algorithms on imprecise domains is mainly due to the following factors:

- **Accuracy:** The standard algorithms are driven by accuracy (minimization of the overall error) to which the minority class contributes very little;
- **Class Distribution:** The current classifiers assume that the algorithms will operate on data drawn from the same distribution as the training data;
- **Error Cost:** The current classifiers assume that the errors coming from different classes have the same costs.

These difficulties are exacerbated by noise, that may further skew class imbalance and be to the learner nearly indistinguishable from rare cases.

Yet, the decision regions induced by a rule-based classifier and the true distribution of the classes in the space of data do not match. Indeed, classes form regions with irregular and interleaved shapes, whereas the induced decision regions are neatly separated by boundaries parallel to the features of the data space. As a consequence, those cases falling within and close to the boundary of a decision region may be misleadingly predicted as belonging to the class associated with that decision region, even if the true class membership in the surroundings of the boundary is different. This is problematic when there is a low separability between classes, i.e. when these form true overlapping (or embedded) regions. In such cases, indeed, the true regions formed by rare classes may be overlapped by the decision regions associated to the predominant classes.

In this chapter, it is decribed a framework that combine associative classification with probabilistic learning [11] to improve classification performance on the rare classes. In imprecise environments, this is preferable with respect to simply increasing classification accuracy, since the latter is strongly biased against rare classes, which as anticipated may also be hardly discriminated from predominant classes. The idea is to use the individual rules of an associative classifier to segment the training data. Segments are used to build as many local probabilistic generative models, that refine the predictions from the corresponding classifier rules. This is particularly useful both in the surroundings of the rule boundaries as well as inside the associated decision regions, wherein local probabilistic generative models act so that classes other than the ones associated to the whole regions influence the classification of nearby unlabeled cases. In practice, local probabilistic models are involved into the classification of unlabeled cases for more effectively dealing with those globally rare cases/classes, that become less rare in the corresponding segments. Two

new schemes for tightly combining associative classification and probabilistic learning are proposed, wherein the class of an unlabeled case is decided by considering multiple class association rules as well as their relative probabilistic generative models. An intensive empirical evaluation shows that, although many possible lines of research for further improvements exist, the hierarchical framework is competitive and often superior in accuracy w.r.t. established competitors, while overcoming them in the ability to deal with rare classes.

5.3 The Hierarchical Predictive Framework

In this section, we discuss our approach to learning a hierarchical framework, that integrates associative classification and local probabilistic generative models. We start with some preliminary notions. Let $\mathcal{D}$ be a relation storing the labeled training cases from which to build an associative classifier. Also, let the schema of $\mathcal{D}$ be a set $\mathcal{A} = \{A_1 : Dom(A_1), \ldots, A_n : Dom(A_n), L : \mathcal{L}\}$ of descriptive attributes. Features $A_1, \ldots, A_n$ are defined over as many categorical or numerical domains, whereas the target class attribute L is a categorical feature. The generic labeled training case $t \in \mathcal{D}$ is a structured tuple, i.e. $t \in Dom(A_1) \times \ldots Dom(A_n) \times Dom(L)$. t can also be equivalently represented in a transactional form. Therein, assume that $\mathcal{M} = \{i_1, \ldots, i_m\}$ is a finite set of items denoting relationships between any attribute of $\mathcal{A}$ but L and a corresponding value. Precisely, the generic item i has the form $A\ [rel]\ v$ where $A \in \mathcal{A} - L$, $v \in Dom(A)$ and $[rel] \in \{=, \leq, \geq\}$ denotes a relationship between A and v. In our formulation, $A = v$ is admissible iff A is a categorical attribute. The remaining relationships $A \leq \tau$ and $A \geq \tau$ are instead allowed iff A is a numeric attribute and, in such a case, τ indicates a generic split point. for a suitable interval of values of A. Split points reflect the discretization of numeric attributes. attributes whose values are symbolic intervals. Any (un)labeled case defined over $\mathcal{A}$ can be modeled as a suitable subset of items in $\mathcal{M}$. Let $\mathcal{L}$ be a finite domain of class labels, the original dataset $\mathcal{D}$ can thus be redefined over $\mathcal{M}$ as a collection $\mathcal{D} = \{t_1, \ldots, t_n\}$ of labeled cases, such that the generic case $t \in 2^{\mathcal{M}} \times \mathcal{L}$. The class label of t is denoted as $class(t)$. Henceforth, we shall adopt the transactional notation.

A class association rule (CAR) $r : I \rightarrow c$ catches an association that occurs in $\mathcal{D}$ between any subset of items $I \subseteq \mathcal{M}$ and a class label $c \in \mathcal{L}$. Notation $class(r)$ represents the class c targeted by r.

The notions of support, coverage and confidence are employed to define the interestingness of a rule r. In particular, A training case $t \in \mathcal{D}$ is said to *support* rule $r : I \rightarrow c$ if it holds that $(I \cup c) \subseteq t$. The support of r is the fraction of training cases supporting r, i.e., $supp(r) = \frac{|\{t \in \mathcal{D} | (I \cup c) \subseteq t\}|}{|\mathcal{D}|}$, where $|\mathcal{D}|$ indicates the cardinality of $\mathcal{D}$.

Rule $r : I \rightarrow c$ is said to *cover* a training case $t \in \mathcal{D}$ (and, dually, t is said to trigger or fire r) if the condition $I \subseteq t$ holds. The set of all training cases covered by r is denoted by $\mathcal{D}_r = \{t \in \mathcal{D} | I \subseteq t\}$. The coverage of r can is the fraction of cases in $\mathcal{D}$ covered by r, i.e. $coverage(r) = \frac{|\mathcal{D}_r|}{|\mathcal{D}|}$. The foresaid rule $r : I \rightarrow c$ is said to *cover* an unlabeled case $I' \subseteq \mathcal{M}$ if $I \subseteq I'$ holds. The confidence of a rule r, denoted by $conf(r)$, is the ratio of support to coverage, i.e. $conf(r) = \frac{supp(r)}{coverage(r)}$.

An associative classifier $\mathcal{C}$ is a suitable disjunction of propositional `if-then` CARs, that predicts the class of an unlabeled case I, i.e. $\mathcal{C}(I) = c \in \mathcal{L}$.

The goal is to learn a hierarchical framework from $\mathcal{D}$, that consists of two classification levels. At the higher level, an associative classifier is built such that its component CARs meet some requirements on the minimum support and confidence. For each CAR $r \in \mathcal{C}$, the lower level of the framework includes a local probabilistic generative model $P^{(r)}$ that allows to confirm or rectify r in the classification of an unlabeled case.

The idea is to build, at the higher level, an associative classifier whose CARs are coupled with local probabilistic generative models, sited at the lower level, that confirm or rectify the predictions from the corresponding CARs. The overall learning process is shown in algorithm 3. Given a database $\mathcal{D}$ of training cases (defined over a set $\mathcal{M}$ of items and a set $\mathcal{L}$ of class labels), the algorithm begins (at line 1) by discovering a set $\mathcal{R}$ of association rules from $\mathcal{D}$ via the MineCARs search strategy. The latter is essentially an enhancement of the Apriori algorithm [1] that integrates multiple minimum class support [74] and complement class support [4] to uncover, within each class, an appropriate number of interesting association rules, whose antecedents and consequents are positively correlated. In particular, within the generic class, multiple minimum class support automatically adjusts the global minimum support threshold σ, provided by the user, to a minimum support threshold specific for that class. Instead, an important property of complement class support is used to retain in $\mathcal{R}$ positively correlated CARs. These are CARs for which the ratio of the observed confidence to the confidence expected by chance (i.e. if the CAR antecedent and consequent were independent) exceeds a class-specific threshold, that is selected without any additional parameter. The exploitation of positively correlated rules allows to overcome a flaw with the support and confidence framework, that produces CARs with poor implicative strength when class distribution is imbalanced, since antecedents and consequents can be negatively correlated [4].

The ruleset $\mathcal{R}$ is then sorted (at line 2) according to the total order $\prec$, which is a refinement of the one introduced in [73]. Precisely, given any two rules $r_i, r_j \in \mathcal{R}$, r_i precedes r_j, which is denoted by $r_i \prec r_j$, if (i) the confidence

Algorithm 3 The hierarchical learning framework

Require: a finite set $\mathcal{M}$ of boolean attributes;
a training dataset $\mathcal{D}$;
a set $\mathcal{L}$ of class labels in $\mathcal{D}$;
and a support threshold σ;
Ensure: An associative classifier $\mathcal{C} = \{r_1 \vee \ldots \vee r_k\}$ and a set of local classifier $\mathcal{P}_{r_i}$;
1: $\mathcal{R} \leftarrow$ MineCARs$(\mathcal{M}, \mathcal{D}, \sigma)$;
2: $\mathcal{R} \leftarrow$ ORDER$(\mathcal{R})$;
3: $\mathcal{C} \leftarrow$ PRUNE$(\mathcal{R})$;
4: **if** there are cases in $\mathcal{D}$ that are not covered by any rule within $\mathcal{C}$ **then**
5: $\quad \mathcal{C} \leftarrow \mathcal{C} \cup \{r_d\}$;
6: **end if**
7: **for** each rule $r \in \mathcal{C}$, such that $r \neq r_d$ **do**
8: $\quad \mathcal{P}^{(r)} \leftarrow$ TrainLocalClassifier(r);
9: **end for**
10: RETURN $\mathcal{C}$ and $\mathcal{P}^{(r)}$ for each $r \in \mathcal{C}$

of r_i is greater than that of r_j, or (ii) their confidences are the same, but the support of r_i is greater than that of r_j, or (iii) both confidences and supports are the same, but r_i is shorter than r_j.

The learning process proceeds (at line 3) to distil a classifier $\mathcal{C}$ by pruning $\mathcal{R}$, which is likely to include a very large number of CARs, that may overfit the training cases. The adopted strategy for overfitting avoidance involves item and rule pruning. Briefly, rule items and/or whole rules are removed from $\mathcal{R}$ whenever this does not worsen the accuracy of the classifier being distilled. The effects of item and rule pruning on the accuracy of the resulting classifier are evaluated using statistical arguments, omitted due to space restrictions. The interested reader is referred to [22] for further details.

The resulting classifier $\mathcal{C}$ may leave some training cases uncovered. Therefore, a default rule $r_d : \emptyset \rightarrow c^*$ is appended to $\mathcal{C}$ (at line 5), such that its antecedent is empty and the targeted class c^* is the majority class among the uncovered training cases.

Finally, for each CAR $r \in \mathcal{C}$ other than the default rule r_d, a local probabilistic model $\mathcal{P}^{(r)}$ is built (lines 7-9) over $\mathcal{D}_r$ to catch a better generalization of those globally rare cases/classes that become less rare within $\mathcal{D}_r$. This allows to refine the prediction from r with a local generative model that is better suited to deal with the local facets of rarity. The TrainLocalClassifier step is treated in the following subsection 5.3.1, that covers the classification of unlabeled cases (not reported in algorithm 3) in the context of two schemes for a tight integration between associative and probabilistic classification.

As a concluding remark, notice that, due to the total order $\prec$ enforced over $\mathcal{R}$, the associative classifier $\mathcal{C}$ is actually a decision list: each training case is classified by the first CAR in $\mathcal{C}$ that covers it. In other words, the CARs in $\mathcal{C}$ are mutually exclusive, i.e. a training case is covered by at most one rule of the classifier. Formally, the definition of the set of training cases covered by the generic CAR $r \in \mathcal{C}$ hereafter becomes $\mathcal{D}_r = \{t \in \mathcal{D} | r \subseteq t \wedge \nexists r' \in \mathcal{C} : r' \prec r, r' \subseteq t\}$. Moreover, the addition to $\mathcal{C}$ (at line 5) of the default rule r_d ensures that the classifier is also exhaustive, i.e. that every training case of $\mathcal{D}$ is covered by at least one CAR of $\mathcal{C}$.

5.3.1 Training Local Classifiers

To improve the predictive accuracy both in the surroundings of decision boundaries as well as within the inner areas of decision regions (wherein classes other than the ones associated to the whole regions may influence the classification of nearby unlabeled cases), each CAR $r \in \mathcal{C}$ is associated with a local probabilistic generative model $\mathcal{P}^{(r)}$, trained over the regularities across the training cases local to $\mathcal{D}_r$. In principle, such regularities are likely to be more descriptive of those globally rare cases/classes that become less rare within $\mathcal{D}_r$. Hence, the individual $\mathcal{P}^{(r)}$ can be involved into the classification process for more accurately dealing with the corresponding forms of rarity.

In the following, we adopt two different probabilistic generative models based, respectively, on the naïve Bayes and nearest neighbor classification models. Precisely, naïve Bayes naturally allows to incorporate the effects of locality on classes and cases in terms of, respectively, class priors and item posteriors. To elucidate, an unlabeled case $I \subseteq \mathcal{M}$ is assigned by the generic generative model $\mathcal{P}^{(r)}$ to the class $c \in \mathcal{L}$ with highest posterior probability

$$\mathcal{P}^{(r)}(c|I) \triangleq p(c|I,r) = \frac{p(I|c,r)p(c|r)}{\sum_{\overline{c} \in \mathcal{L}} p(I|\overline{c},r)p(\overline{c}|r)} = \frac{\prod_{i \in I} p(i|c,r)p(c|r)}{\sum_{\overline{c} \in \mathcal{L}} \prod_{i \in I} p(i|\overline{c},r)p(\overline{c}|r)}$$

Locality influences factors $p(c|r)$'s and $p(i|c,r)$'s, whose values are estimated by computing $p(c)$ and $p(i|c)$ over $\mathcal{D}_r$, and allows to better value rare cases/classes. Indeed, if a significant extent of some form of rarity falls within $\mathcal{D}_r$, the corresponding cases/classes are obviously less rare than in $\mathcal{D}$ and, hence, factors $p(c)$'s and $p(i|c)$'s are accordingly higher (w.r.t. their values in $\mathcal{D}$). Dually, $p(c)$'s and $p(i|c)$'s are sensibly lower, if the density of that form of rarity within $\mathcal{D}_r$ is much lower than in $\mathcal{D}$. However, this is acceptable, since most of that form of rarity is still captured within some other region(s). An inconvenient behind the adoption of naïve Bayes as the underlying model for local probabilistic classifiers is their performance degrade (e.g. accuracy loss) due to the violation of the attribute independence assumption. To alleviate

such an issue, the weaker attribute independence assumption postulated in AODE [115] can be plugged into the above formulation, that simply refines naïve Bayes by considering each attribute dependent upon at most n other attributes in addition to the class. This is more realistic in practice and is empirically shown in section 5.4 to yield a better performance.

Another difficulty behind naïve Bayes is that the estimates of some class priors and item posteriors may not be reliable when data is too rare within $\mathcal{D}_r$. In such cases, the nearest neighbor model can be alternatively used to compute probabilities $\mathcal{P}^{(r)}(c|I)$ from the distribution of classes within $\mathcal{D}_r$ through the generative approach below

$$\mathcal{P}^{(r)}(c|I) \triangleq \frac{\sum_{I' \in \mathcal{D}_r} w_{I'} p(c|I')}{\sum_{\overline{c} \in \mathcal{L}} \sum_{I' \in \mathcal{D}_r} w_{I'} p(\overline{c}|I')}$$

The above is essentially a probabilistic re-formulation of a distance-weighted voting scheme, in which each neighbor I' votes for the class that should be assigned to I. The vote from the generic neighbor I' is suitably weighted by a corresponding factor $w_{I'}^{(r)}$, which takes into account the actual distance between I' and I. Formally,

$$w_{I'} = \frac{e^{-d^2(I,I')}}{\sum_{I' \in \mathcal{D}_r} e^{-d^2(I,I')}}$$

where $d(I, I')$ is any suitable function that defines a notion of distance between I and I'. Notice that, whatever the distance between cases, the chosen weight-definition attributes higher influences to those neighbors in $\mathcal{D}_r$ that are actually closest to I.

Two alternative approaches for refining the predictions from the associative classifier $\mathcal{C}$ through the local probabilistic generative models $\mathcal{P}^{(r)}$'s are discussed next.

Local priors and local instance posteriors.

The idea is to reformulate a generative approach to classification which spans into local generative models. Starting from the observation that the exhaustive and exclusive rules within $\mathcal{C}$ partition the space of covering events relative to

a tuple, it is possible to define the joint probability over unlabeled cases and a class labels as shown below

$$p(c, I) = \sum_{r \in \mathcal{C}} p(c, I, r) = \sum_{r \in \mathcal{C}} p(c, I|r)p(r) = \sum_{r \in \mathcal{C}} \mathcal{P}^{(r)}(c|I)p(I|r)p(r)$$

Within the above formula, $p(I|r)$ represents the compatibility of I with the rule r. We choose to model $p(I|r)$ as the relative number of items that I shares with r: intuitively, the number of (mis)matches represents the closeness of I to the region bounded by r. $\mathcal{P}^{(r)}(c|I)$ denotes the probability associated with c by the local naïve Bayes classifier $\mathcal{P}^{(r)}$ trained over $\mathcal{D}_r$. $p(r)$ indicates the support $supp(r)$ of CAR r and weights its contributions to $p(c, I)$ by the relative degree of rarity of its antecedent and consequent.

Finally, the probability of class c given the unlabeled case I can be formalized as the following generative model

$$p(c|I) = \frac{p(c, I)}{\sum_{\overline{c} \in \mathcal{L}} p(\overline{c}, I)}$$

Cumulative rule effect.

A stronger type of interaction between global and local effects can be injected into the classification process, if the predictions from a CAR r and unrelated local generative model $\mathcal{P}^{(r')}$ (with $r \neq r'$) are compared for selecting the most confident one. The overall approach sketched in fig. 4. Precisely, the generic unlabeled case $I \subseteq \mathcal{M}$ is presented to the associative classifier $\mathcal{C}$ and the first CAR $r : I \rightarrow c$ (in the precedence order $\prec$ enforced over $\mathcal{C}$) is chosen (at line 1). If r does not cover I, it is skipped and the next rule is recursively taken into account (at line 20). Otherwise, r is used for prediction. However, its target class c is not directly assigned to I. Rather, the local probabilistic generative model $\mathcal{P}^{(r)}$ corresponding to r is exploited to produce a possibly more accurate prediction (at line 4). Some tests are performed to identify the more confident prediction (lines 9- 15). If both counterparts agree or one is deemed to be more reliable than the other one, the better prediction (in terms of class-membership probability distribution) is returned (lines 10 and 12). Otherwise, in the absence of strong evidence to reject the prediction from $\mathcal{P}^{(r)}$ (which is in principle preferable to r, being more representative of the local regularities that may come from globally rare cases/classes that fall within $\mathcal{D}_r$), r is skipped in favor of the next CAR $r' \in \mathcal{C}$ covering I (at line 14). To this point, if $\mathcal{P}^{(r')}$ predicts I more confidently than $\mathcal{P}^{(r)}$ (at line 5), the probability distribution from $\mathcal{P}^{(r')}$ replaces the current best distribution yielded by $\mathcal{P}^{(r)}$ (at line 6) and the choice of a better prediction is hence made between r' and $\mathcal{P}^{(r')}$. In the opposite case, the choice involves r'

and the current best distribution $\mathcal{P}^{(r)}$. If no prediction is clearly eligible as the most confident throughout the search, the process halts when the default rule is met and the current best distribution is returned (at line 17). Notice that the sofar best class-membership probability distribution is remembered throughout the consecutive stages of the search process via the input arguments $p_1, \ldots, p_k$ (such arguments are individually set to 0 at the beginning of the search process). A key aspect of the overall search process is represented by the criteria adopted to choose the more confident prediction between the ones from a CAR r_h and a local probabilistic generative model $\mathcal{P}^{(r_i)}$. Accuracy is used as a discriminant between the alternatives. In particular, the accuracy $acc^{(c)}\left(\mathcal{P}^{(r_i)}\right)$ is the percent of cases in $\mathcal{D}^{(r)}$ correctly predicted by $\mathcal{P}^{(r_i)}$ as belonging to class c. The accuracy $acc^{(c)}(r_h)$ of a CAR r_h predicting class c is its confidence $conf(r_h)$. When comparing the accuracies of a CAR r_h and a local probabilistic generative model $\mathcal{P}^{(r_i)}$ there are four possible outcomes.

1. $\mathcal{P}^{(r_i)}$ is clearly deemed more reliable than r_h (at line 9), if the weighted accuracy of the former, p^*, is greater than the accuracy of the latter.

2. r_h is preferred to $\mathcal{P}^{(r_i)}$ (at line 11) if the accuracy of the former is greater than or equal to the weighted accuracy of the latter and both agree anyhow.

3. r_h is preferred to $\mathcal{P}^{(r_i)}$ (again at line 11) if its accuracy is much greater than the weighted accuracy of $\mathcal{P}^{(r_i)}$. Therein, $\frac{p^*}{p} > p*$ is a prudential threshold, that represents the normalized weighted accuracy from $\mathcal{P}^{(r_i)}$. In practice, r_h is actually preferable to $\mathcal{P}^{(r_i)}$ iff its accuracy exceeds $\frac{p^*}{p}$.

4. There is no strong evidence (at line 16) to reject either r_h or $\mathcal{P}^{(r_i)}$ when the accuracy of r_h lies in the interval $(p*, \frac{p^*}{p})$. In such a case, r is skipped and the search proceeds to considering the next CAR in the associative classifier $\mathcal{C}$ that covers I (through the recursive call at line 14).

5.4 Evaluation

It is evaluated experimentally the behavior of the hierarchical classification framework to understand whether it exhibits improvements in classification performance with respect to established competitors. For the comparative evaluation, we use some standard datasets from the UCI KDD repository [5] with high class imbalance. Tests are performed over two further datasets. `kdd99` is the KDD99 intrusion detection dataset, wherein class distribution is strongly skewed and low-frequency classes are affected by noise. `fraud` is a

Algorithm 4 The scheme for classifying an unlabeled case under the cumulative rule effect

Require: An associative classifier $\mathcal{C}$;
an unlabeled case $I \subseteq \mathcal{M}$;
Ensure: the class distribution for I;
select the first rule $r : I' \rightarrow c_h$ in sequence within $\mathcal{C}$;
if r covers I (i.e. $I' \subseteq I$) **then**
  if $|\mathcal{C}| > 1$ (i.e. r is not the default rule) **then**
    let $\overline{p}_i = \mathcal{P}^{(r)}(c_i|I) \cdot acc^{(c_i)}\left(\mathcal{P}^{(r)}\right),\ \forall i = 1, \ldots, k$;
    if $max_i(\overline{p}_i) > max_i(p_i)$ **then**
      let $p_i = \overline{p}_i,\ \forall i = 1, \ldots, k$;
    end if
    let $p^* = max_i(p_i)$ and $i^* = argmax_i(p_i)$ and $p = \sum_i p_i$;
    if $acc^{(c_h)}(r) < p^*$ **then**
      RETURN the distribution $(p_1/p, \ldots, p_k/p)$;
    else if $i^* = h$ **or** $acc^{(c_h)}(r) > \frac{p^*}{p}$ **then**
      RETURN the distribution $(acc^{(c_1)}(r), \ldots, acc^{(c_k)}(r))$;
    else
      Prediction($\mathcal{C} - \{r\}$,I,$p_1, \ldots, p_k$);
    end if
  else
    RETURN the distribution $(p_1/p, \ldots, p_k/p)$;
  end if
else
  Prediction($\mathcal{C} - \{r\}$,I,$p_1, \ldots, p_k$);
end if

(non-publicly available) real-life fraud detection dataset, with a very low class separability.

It is remarked that, as pointed out in [117], the effectiveness of a classification strategy on rare cases cannot be directly evaluated, since these are usually unknown. Notwithstanding, both rare classes and rare cases are argued to be two strongly related facets of rarity, whose issues can be addressed with the same methods. Hence, we expect that if an approach is effective with rare classes, it is also useful for dealing with rare cases. Experiments consists in comparisons against several established rule-based and associative classifiers. The selected rule-based competitors are Ripper [30] and PART [55], while the associative ones include CBA [73] and CMAR [125]. In particular, we exploited the implementations of CBA and CMAR in [29].

Numeric attributes in the chosen datasets are discretized for all schemes but Ripper, through equal-frequency binning. Moreover, the test involving CBA and CMAR are reiterated several times, under different settings for the

minimum support and confidence parameters: we next report the results corresponding to the best parameter configuration allowed by the implementations at [29]. Overall, the results from the individual classifiers were averaged over ten-fold cross-validation.

The proposed schemes simply require the specification of a global minimum support. Due to the adoption of minimum class support [73], such threshold is automatically adjusted to become a class specific threshold. In particular, we fixed the global support threshold to 20%, which is transparently adjusted to be, within the individual class in the data at hand, the 20% of the frequency of that class. The exploitation of complement class support [74] permits to avoid specifying a minimum confidence threshold.

We compare the approaches using accuracy, some meaningful ROC curves and the Area Under the Curve (AUC) relative to the minority class. table 5.1 and table 5.2 display the results. Within the tables, (1) indicates Ripper, (2) corresponds to PART, while (3) and (4) stand for CBA and CMAR, respectively. Our schemes are instead numbered from (5) to (10). More specifically, (5) and (6) indicate naive Bayesian smoothing (respectively through local priors or cumulative effect). (7) and (8) stand for nearest-neighbor smoothing (respectively, through local priors or cumulative effect). (9) and (10) are AODE smoothing (respectively, through local priors or cumulative effect).

Dataset	**(1)**	**(2)**	**(3)**	**(4)**	**(5)**	**(6)**	**(7)**	**(8)**	**(9)**	**(10)**
anneal	96.63	96.66	92.81	96.33	96.76	96.76	96.70	96.63	96.76	96.76
balance-scale	77.29	77.27	68.81	68.49	77.84	77.53	74.04	77.60	**79.67**	78.01
breast-cancer	71.46	68.54	69.20	67.67	70.52	70.52	72.65	71.92	72.44	72.44
horse-colic	84.26	81.95	81.62	83.96	82.42	82.42	84.12	83.85	82.11	82.11
credit-rating	86.28	85.07	81.74	83.76	85.78	85.78	**86.57**	86.43	86.06	86.06
german-credit	71.74	72.24	73.10	73.34	74.21	74.21	72.48	71.85	**74.53**	74.53
pima-diabetes	77.41	76.84	77.87	73.03	78.06	78.06	77.97	76.86	77.63	77.63
glass	71.95	74.94	72.69	74.23	75.63	74.93	73.72	72.94	**76.66**	75.65
cleveland-14-heart-disease	82.24	80.46	82.12	75.12	81.78	81.97	82.43	82.20	82.34	82.33
hungarian-14-heart-disease	80.48	81.29	82.06	79.69	82.87	82.90	81.60	81.67	**83.00**	82.97
heart-statlog	82.89	83.33	82.59	84.19	83.34	84.19	82.74	81.93	84.52	84.52
hepatitis	80.58	78.20	79.89	81.08	**82.17**	81.08	80.38	80.19	80.85	80.85
ionosphere	91.68	90.03	87.89	89.74	**93.72**	89.74	92.28	92.28	92.85	92.85
labor	83.33	84.63	86.67	88.77	87.17	88.77	83.33	83.33	88.23	88.23
lymphography	79.14	80.20	81.18	80.59	**84.16**	80.45	79.68	79.54	80.21	80.21
sick	97.60	97.87	97.51	97.64	93.88	97.64	97.51	97.57	97.65	97.65
sonar	79.00	81.26	80.00	82.78	63.36	82.78	80.10	79.67	82.64	82.64
fraud	93.07	93.02	80.82	90.52	91.79	91.79	**93.05**	92.96	92.61	92.61
kdd99	96.61	96.98	94.65	94.63	95.98	95.98	96.78	96.73	96.65	96.65

Table 5.1: Classification accuracy

Dataset	(1)	(2)	(5)	(6)	(7)	(8)	(9)	(10)
anneal	0.79	0.91	0.94	0.94	0.93	0.93	0.95	0.95
balance-scale	0.82	0.90	0.94	0.92	0.93	0.92	0.94	0.92
breast-cancer	0.60	0.59	0.68	0.68	0.63	0.63	0.69	0.69
horse-colic	0.83	0.83	0.85	0.85	0.87	0.87	0.87	0.87
credit-rating	0.87	0.91	0.92	0.92	0.91	0.91	0.93	0.93
german-credit	0.63	0.71	0.77	0.77	0.72	0.72	0.78	0.78
pima-diabetes	0.75	0.81	0.84	0.84	0.83	0.82	0.84	0.84
glass	0.87	0.89	0.87	0.85	0.87	0.85	0.87	0.86
cleveland-14-heart-disease	0.83	0.84	0.90	0.90	0.89	0.88	0.90	0.90
hungarian-14-heart-disease	0.78	0.86	0.90	0.90	0.89	0.89	0.90	0.90
heart-statlog	0.83	0.85	0.90	0.90	0.88	0.87	0.90	0.90
hepatitis	0.70	0.69	**0.84**	**0.84**	0.78	0.77	**0.84**	**0.84**
ionosphere	0.92	0.92	0.95	0.95	0.94	0.94	0.98	0.98
labor	0.81	0.83	0.96	0.96	0.90	0.90	0.96	0.96
lymphography	0.46	0.56	**0.99**	0.79	0.81	0.68	0.97	0.92
sick	0.91	0.93	0.96	0.96	0.96	0.96	0.96	0.96
sonar	0.81	0.86	0.92	0.92	0.90	0.89	0.92	0.92
fraud	0.68	0.77	0.81	0.81	0.78	0.78	**0.92**	0.90
kdd99	0.98	0.99	0.99	0.99	0.99	0.99	0.99	0.99

Table 5.2: Area Under the Curve

The results clearly state that the combination of associative classification and probabilistic smoothing is at least as accurate as the seminal rule-based classifiers chosen for the comparison. In many cases, however, (5) and (11) achieve improvements in accuracy, reported in bold within table 5.1, that are statistically significant according to the t-test. In addition, a deeper analysis reveals that the response versus the classes of interest is strongly improved. Such an improvement can be appreciated by looking at the details of the individual datasets. We report in table 5.3 the confusion matrices originated by (1) (6) and (9) over the `german-credit` dataset: the probabilistic smoothing here recovers 39 tuples to the minority class, thus allowing to achieve a higher precision.

Predicted ->	good	bad
good	607	93
bad	155	145

AODE local priors (9)

Predicted ->	good	bad
good	611	89
bad	194	106

Ripper (1)

Table 5.3: The confusion matrices yielded by AODE local priors (9) and Ripper (1)

A further analysis of the results obtained over the `fraud` and the `kdd99` datasets provides an in-depth into the effects of smoothing. The figure 5.1 shows the ROC curves relative to (1), (2), (5), (7) and (9). There is an ev-

ident improvement in the underlying area with respect to the competitors (1) and (2), whose trends are plotted in red. Results with the `kdd99` dataset are even more surprising, and in particular with the `u2r` class, as shown in figure 5.2, that represents the curves relative to the schemes (1), (2) and (9). The `u2r` class is made of 56 tuples (out of 150K), and still the probabilistic adjustment is capable of recovering some problematic cases.

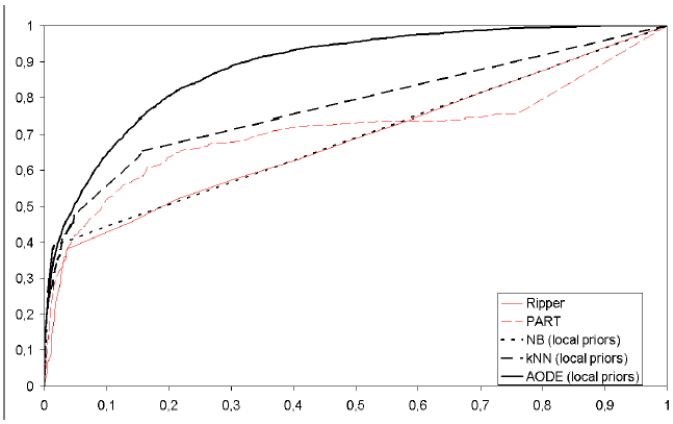


Fig. 5.1: ROC curve for the minority class in the `fraud` dataset

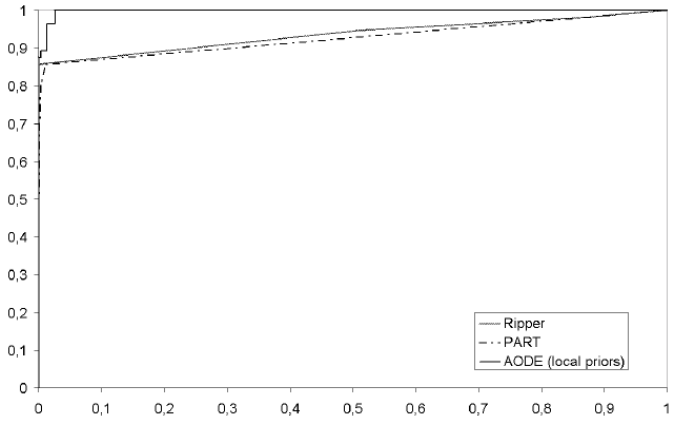


Fig. 5.2: ROC curve for the minority `u2r` class in the `kdd99` dataset

Finally, the ability of the approaches at dealing with the classes is compared in table 5.2, which reports the AUC values across the selected datasets. The AUC is a measure of the separability of two classes. Therefore, for the two-class datasets, table 5.2 simply reports a scalar value, which indicates the capability of the classification schemes at discriminating between the positive (i.e. rare) and negative (i.e. predominant) classes. For the multi-class datasets, table 5.2 combines multiple pairwise separability values by following the class reference approach in [49] and reports the weighted sum of the resulting pairwise AUC values (weights are the occurrence frequencies of each reference class). The devised schemes exhibit an improved performance across all classes within the distinct datasets and, in particular, with `hepatitis`, `lymphography` and `fraud`, where the improvement is over 10%. As witnessed by the graphs in figure 5.1 and figure 5.2, such an overall improvement is primarily obtained on the minority classes.

6

Conclusion and future research

The main focus of this thesis was *Data Mining in Fraud Detection*, in other words, on the use of the best data mining methods and techniques to predict fraudolent behaviours in enviroment featured by large and highly skewed data.

For this purpose, this dissertation first explored existing fraud detection methods and evaluated the supervised, unsupervised, score-based, and rule-based approaches to determine which ones are more important. Then several methods and techniques are proposed in order to improve the model quality in imprecise domains and to identify "exceptional" fraudolent behaviours.

The first contribution was Sniper, a predictive modeling technique for multi-purpose fiscal fraud detection in presence of biased and unbalanced training sets. The methodology produces a rule-based classification system that can be tuned to the requirements of the auditing agency, since it concentrates on a user-defined fixed number of most prominent subjects recognizable as fraudsters. The methodology was succesfully applied, in collaboration with the Italian Revenue Agency, to the case of VAT refund fraud, although it can be generalized to other situations where fraud detection can be characterized by a multi-purpose objective in presence of a noisy environment.

The Sniper methodology is currently being validated on real cases: a number of subjects have been selected on the basis of the Sniper rules, and actual audits are being performed, in order to asses the predictive accuracy and effectiveness.

Two relevant outcomes are currently substantiating in the validation process. The first is that most of the audited subjects are unexpected cases, i.e., subjects the experts should have never selected for auditing based on their current practices. That is, the adoption of Data Mining methodology can ease the discovery of new fraud behaviors. The second result is that audited subjects found positive typically met all the three criteria of proficiency, equity

and efficiency. Proficiency and efficiency exhibit values close to those in the top class in the training set. Equity, by contrast, exhibit better values, with increases ranging from 1% to 37%. The meaning is that the model succeeds in pursuing a multi-purpose objective, being in particular able to identify subjects with high fraud with respect to business volume. Since these subjects were generally ignored by current audit practices, the Sniper methodology may represent a significant advance in strategic planning for fiscal fraud detection.

The second contribution was a new technique, named Numeric SNIPER , that deals with the problem of learning a model of object exceptionality from a collection of training objects for which a continuous measure of their exceptionality is known. The resulting model is then useful for the identification of exceptionality within any further collection of objects, whose extent of exceptionality is unknown. Numeric SNIPER relies on novel contributions regarding the redefinition of the notions of rule accuracy and entropy with respect to a continuous target attribute. A preliminary comparative evaluation revealed that, although there are still opportunities for further improvements, Numeric SNIPER is competitive and often outperforming in terms of both result quality and computational efficiency.

The ongoing research efforts are geared towards the study of more sophisticated notions of rule accuracy, that better guide the search for dispersed nuggets of highly exceptional objects with low variability in their extent of exceptionality. The impact of such notions on the behavior of Numeric SNIPER shall also be empirically evaluated on a variety of critical datasets (wherein unexceptional objects overwhelm the exceptional ones) against a larger number of heterogeneous competitors.

Future research aim to improve the accuracy of the local probabilistic generative models through ROC analysis [11]. The classification threshold used in our framework assigns a class label when the associated probability is higher than 0.5. However, the latter may not necessarily be the best threshold, especially if we consider the bias introduced by the CAR associated with the probabilistic classifier. In general, higher thresholds produce improvements in recall, by contemporarily degrading precision. However, by automatically selecting the best class-specific threshold, probabilistic smoothing can still allow to remove some locality effects within the CAR and maintain high precision as well.

Also, we intend to investigate better strategies for overfitting avoidance into the scheme of algorithm 3. The currently adopted pruning method aims to improve the classification accuracy of the resulting CARs and, hence, can introduce a bias in favor of majority classes, that would likely lead to clas- sifiers predicting very poorly the rare classes [119]. To overcome such a limitation,

the idea is to choose a compact subset of the original CARs, that maximizes the area under the ROC curve for each class in the training data. This would decouple classification performance from class imbalance and properly account for rare classes.

Finally, we aim at carrying forward the study of CARs, whose antecedents are maximal item associations. The purpose is relaxing the global independence assumption behind Bayes theorem (that involves all case items). A new local independence assumption, which only focuses on those items of an unlabeled case, that do not appear in the antecedent of the CAR covering the case, is more realistic in practical applications.

A

Appendix A

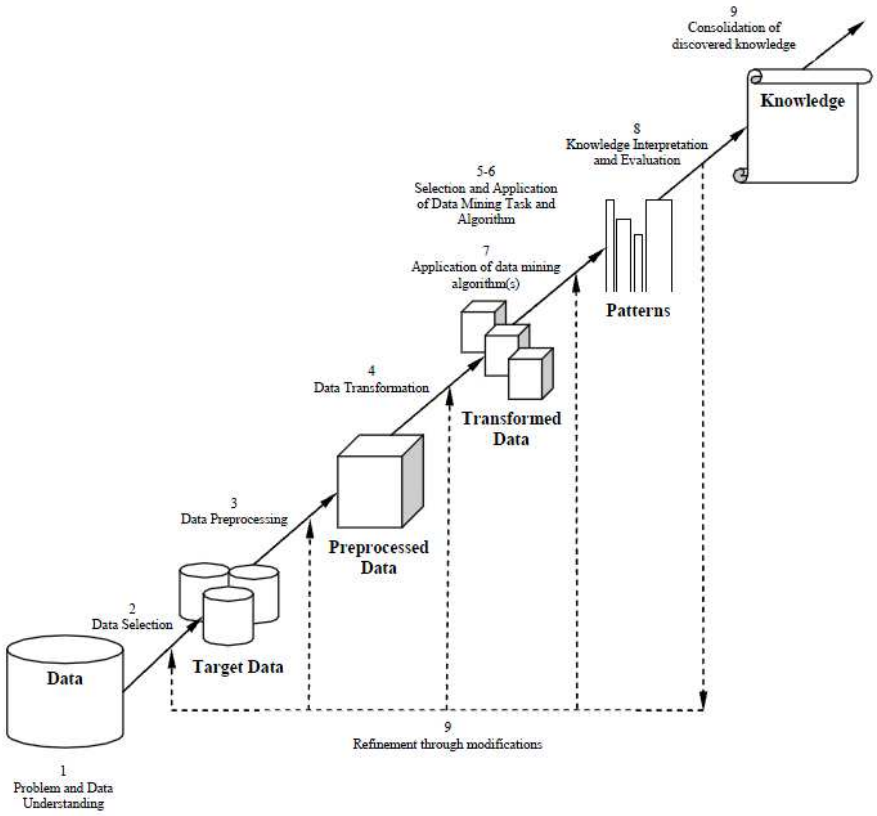


Fig. A.1: GENERIC DATA MINING PROCESS

B

Appendix B

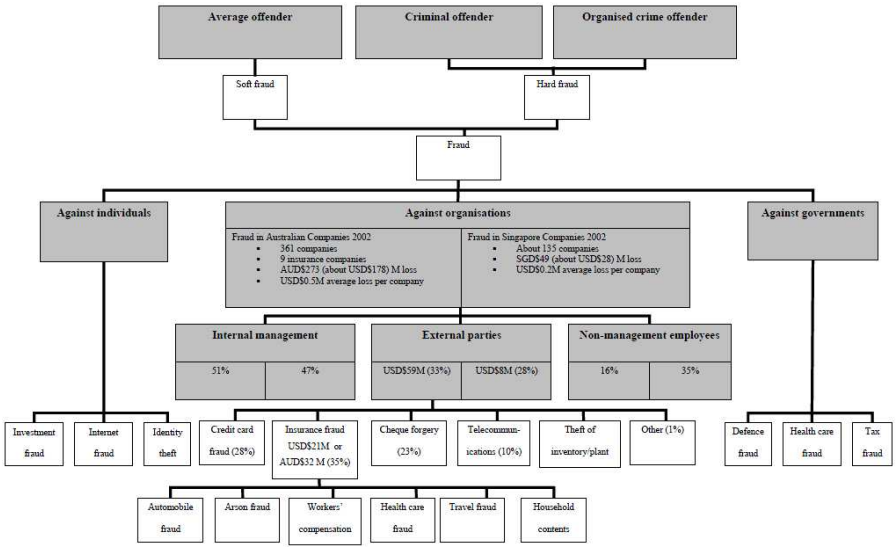
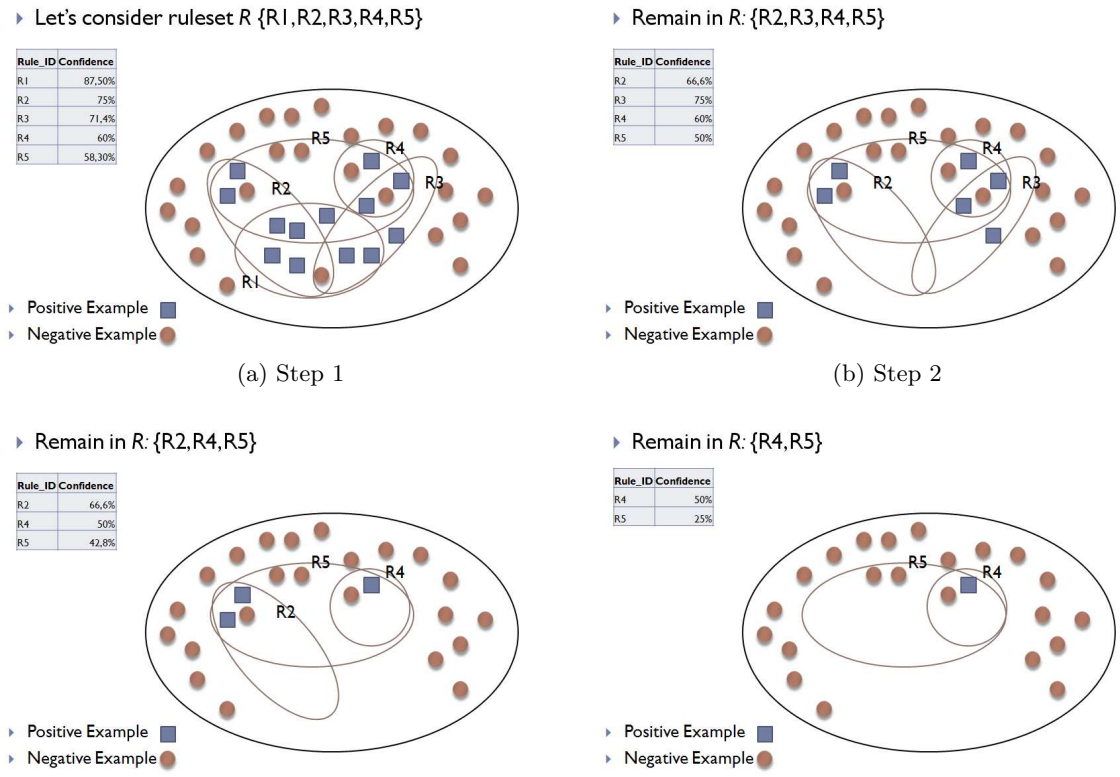


Fig. B.1: TAXONOMY OF FRAUD

C

Appendix C



(a) Step 1

(b) Step 2

(c) Step 3

(d) Step 4

Fig. C.1: MERGING RULES EXAMPLE

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. of Int. Conf. on Very Large Data Bases*, pages 487 – 499, 1994.
2. Emin Aleskerov, Bernd Freisleben, and Bharat Rao. CARDWATCH: A neural network based database mining system for credit card fraud detection. In *Computational Intelligence for Financial Engineering, Proceedings of the IEEE/IAFE*, pages 220–226, 1997.
3. M.-L. Antonie and O.R. Zaïane. Text document categorization by term association. In *Proc. of IEEE Int. Conf. on Data Mining*, pages 19–26, 2002.
4. B. Arunasalam and S. Chawla. CCCS: A top-down association classifier for imbalanced class distribution. In *Proc. of ACM KDD*, pages 517–522, 2006.
5. A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.
6. Emilie Lundin Barse, Håkan Kvarnström, and Erland Jonsson. Synthesizing test data for fraud detection systems. In *Proceedings of the 19th Annual Computer Security Applications Conference*, ACSAC '03, pages 384–, Washington, DC, USA, 2003. IEEE Computer Society.
7. S. Basta and Others. High-quality true positive prediction for fiscal fraud detection. Technical Report 01/09, ICAR-CNR, 2009. Available at `http://www.icar.cnr.it/basta/TR12009.pdf`.
8. Peter J. Bentley. "Evolutionary, my dear Watson" Investigating Committee-based Evolution of Fuzzy Rules for the Detection of Suspicious Insurance Claims. In *Genetic and Evolutionary Computation Conf.(GECCO-2000)*, pages 702–709, 2000.
9. Peter J. Bentley, Jungwon Kim, Gil H. Jung, and Jong U. Choi. Fuzzy Darwinian Detection of Credit Card Fraud. In *14th Annual Falll Symposium of the Korean Information Processing Society*, 2000.
10. Michael Berry and Gordon Linoff. *Mastering Data Mining: The Art and Science of Customer Relationship Management*. John Wiley & Sons, Inc., New York, NY, USA, 1st edition, 1999.
11. C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.
12. Richard J. Bolton and David J. Hand. Unsupervised Profiling Methods for Fraud Detection. *Statistical Science*, 17(3):235–255, 2002.
13. F. Bonchi, F. Giannotti, G. Mainetto, and D. Pedreschi. A classification-based methodology for planning audit strategies in fraud detection. In *Proceedings of*

the fifth ACM SIGKDD international conference on Knowledge discovery and data mining, KDD '99, pages 175–184, New York, NY, USA, 1999. ACM.

14. R. Brause, T. Langsdorf, and M. Hepp. Neural data mining for credit card fraud detection. In *Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence*, ICTAI '99, pages 103–, Washington, DC, USA, 1999. IEEE Computer Society.
15. Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.
16. Patrick L. Brockett, Richard A. Derrig, Linda L. Golden, Arnold Levine, and Mark Alpert. Fraud classification using principal component analysis of ridits, 2002.
17. Peter Burge and John Shawe-Taylor. An unsupervised neural network approach to profiling the behavior of mobile phone users for use in fraud detection. *J. Parallel Distrib. Comput.*, 61:915–925, July 2001.
18. Peter Cabena, Pablo Hadjinian, Rolf Stadler, Jaap Verhees, and Alessandro Zanasi. *Discovering data mining: from concept to implementation*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1998.
19. M. Cahill, D. Lambert, J. Pinheiro, and D. Sun. Detecting fraud in the real world, 2000.
20. Michael H. Cahill, Diane Lambert, José C. Pinheiro, and Don X. Sun. Chapter 1 DETECTING FRAUD IN THE REAL WORLD.
21. Michael H. Cahill, Diane Lambert, José C. Pinheiro, and Don X. Sun. Handbook of massive data sets. chapter Detecting fraud in the real world, pages 911–929. Kluwer Academic Publishers, Norwell, MA, USA, 2002.
22. E. Cesario, F. Folino, A. Locane, G. Manco, and R. Ortale. Boosting text segmentation via progressive classification. *Knowledge and Information Systems*, 15(3):285–320, 2008.
23. Philip K. Chan, Wei Fan, Andreas L. Prodromidis, and Salvatore J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, 14:67–74, 1999.
24. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.
25. Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.
26. Rong-Chang Chen, Ming-Li Chiu, Ya-Li Huang, and Lin-Ti Chen. Detecting credit card fraud by using questionnaire-responded transaction model based on support vector machines. In Zheng Rong Yang, Richard M. Everson, and Hujun Yin, editors, *IDEAL*, volume 3177 of *Lecture Notes in Computer Science*, pages 800–806. Springer, 2004.
27. Chuang-Cheng Chiu and Chieh-Yuan Tsai. A web services-based collaborative scheme for credit card fraud detection. *e-Technology, e-Commerce, and e-Services, IEEE International Conference on*, 0:177–181, 2004.
28. Graham J. Williams (cmis/cbr, Zhexue Huang (cmis/cbr, Graham J. Williams, and Zhexue Huang. Mining the knowledge mine: The hot spots methodology for mining large real world databases, 1997.
29. F. Coenen. LUCS KDD implementations of CBA and CMAR, 2004.
30. W. W. Cohen. Fast effective rule induction. In *Proc. of Conf. on Machine Learning*, pages 115–123, 1995.

31. William W. Cohen. Fast effective rule induction. In *Procs of ICML*, pages 115–123, 1995.
32. Converium. Tackling insurance fraud - law and practice, 2002.
33. Corinna Cortes and Daryl Pregibon. Signature-based methods for data streams. *Data Min. Knowl. Discov.*, 5:167–182, July 2001.
34. Corinna Cortes, Daryl Pregibon, Chris Volinsky, and At&t Shannon Labs. Computational methods for dynamic graphs. *Journal of Computational and Graphical Statistics*, 12:950–970, 2003.
35. Kenneth C. Cox, Stephen G. Eick, Graham J. Wills, and Ronald J. Brachman. Visual data mining: Recognizing telephone calling fraud, 1997.
36. Ghosh S. & Reilly D. Credit card fraud detection with a neural network, 1994.
37. Major J. & Riedinger D. A hybrid knowledge/statistical-based system for the detection of fraud, 2002.
38. Carlos Domingo, Ricard Gavaldà, and Osamu Watanabe. Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Min. Knowl. Discov.*, 6:131–152, April 2002.
39. P. Domingos. Metacost: A general method for making classifiers cost-sensitive. In *Procs. of KDD*, pages 155–164, 1999.
40. José R. Dorronsoro, Francisco Ginel, Carmen Sánchez, and Carlos S. Cruz. Neural Fraud Detection in Credit Card Operations. *IEEE Transactions On Neural Networks*, 8(4):827–834, July 1997.
41. Chris Drummond and Robert C. Holte. C4.5, Class Imbalance, and Cost Sensitivity: Why Under-sampling beats Over-sampling, 2003.
42. Charles Elkan. Magical thinking in data mining: Lessons from coil challenge 2000. In *In Knowledge Discovery and Data Mining*, pages 426–431, 2001.
43. Vladimir Estivill-Castro and Ickjai Lee. Data mining techniques for autonomous exploration of large volumes of geo-referenced crime data. In *Proc. of the 6th International Conference on Geocomputation*, 2001.
44. K. Ezawa, M. Singh, and S.W. Norton. Learning goal oriented bayesian networks for telecommunications risk management. In *Proc. of Int. Conf. on Machine Learning*, pages 139–147, 1996.
45. Kazuo J. Ezawa and Steven W. Norton. Constructing bayesian networks to predict uncollectible telecommunications accounts. *IEEE Expert: Intelligent Systems and Their Applications*, 11:45–51, October 1996.
46. Belhadji E. Dionne G. & Tarkhani F. A model for the detection of insurance fraud, 2000.
47. Stefano B. & Gisella F. Insurance fraud evaluation: A fuzzy expert system., 2001.
48. Wei Fan. Systematic data selection to mine concept-drifting data streams. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '04, pages 128–137, New York, NY, USA, 2004. ACM.
49. T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27(8):861–874, 2006.
50. Tom Fawcett. Ai approaches to fraud detection and risk management: Papers from the 1997 aaai workshop. Technical Report WS-97-07, AAAI Press, 1997.
51. Tom Fawcett. "in vivo" spam filtering: a challenge problem for kdd. *SIGKDD Explor. Newsl.*, 5:140–148, December 2003.

52. Tom Fawcett and Foster Provost. Activity monitoring: Noticing interesting changes in behavior. In Chaudhuri and Madigan, editors, *Proceedings on the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 53–62, San Diego, CA, August 1999.
53. Usama M. Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. From data mining to knowledge discovery: An overview. In *Advances in Knowledge Discovery and Data Mining*, pages 1–34. AAAI Press, 1996.
54. Dean P. Foster and Robert A. Stine. Variable selection in data mining: Building a predictive model for bankruptcy. *Journal of the American Statistical Association*, pages 303–313, 2004.
55. E. Frank and I. H. Witten. Generating accurate rule sets without global optimization. In *Proc. of Int. Conf. on Machine Learning*, pages 144–151, 1998.
56. J. Han and Y. Yin. Mining frequent patterns without candidate generation. In *Proc. of ACM SIGMOD Int. Conf. on Management of data*, pages 1–12, 2000.
57. Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2000.
58. David J. Hand. Strength in diversity: The advance of data analysis. In Jean-François Boulicaut, Floriana Esposito, Fosca Giannotti, and Dino Pedreschi, editors, *PKDD*, volume 3202 of *Lecture Notes in Computer Science*, pages 18–26. Springer, 2004.
59. Hongxing He, Warwick Graco, and Xin Yao. Application of genetic algorithm and k-nearest neighbour method in medical fraud detection. In *Selected papers from the Second Asia-Pacific Conference on Simulated Evolution and Learning on Simulated Evolution and Learning*, SEAL'98, pages 74–81, London, UK, 1999. Springer-Verlag.
60. Victoria Hodge and Jim Austin. A Survey of Outlier Detection Methodologies. *Artificial Intelligence Review*, 22(2):85–126, October 2004.
61. R. V. Hogg and A. T. Craig. *Introduction to Mathematical Statistics*. Prentice Hall, 1995.
62. Jaakko Hollmn and Volker Tresp. Call-based fraud detection in mobile communication networks using a hierarchical regime-switching model. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 889–895. MIT Press, 1999.
63. G. Holmes, M. Hall, and E. Frank. Generating rule sets from model trees. In *Proc. of Australian Joint Conf. on Artificial Intelligence*, pages 1–12, 1999.
64. R.C. Holte, L.E. Acker, and B.W. Porter. Concept learning and the problem of small disjuncts. In *Procs. 11th IJCAI Conf.*, pages 813–818, 1989.
65. Hyun-Chul Kim, Shaoning Pang, Hong-Mo Je, Daijin Kim, and Yang. Constructing support vector machine ensemble. *Pattern Recognition*, 36(12):2757–2767, December 2003.
66. Jungwon Kim, A. Ong, and R. E. Overill. Design of an artificial immune system as a novel anomaly detector for combating financial fraud in the retail sector. In *CEC '03: The 2003 Congress on Evolutionary Computation*, volume 1, pages 405–412, December 2003.
67. A. I. Kokkinaki. On atypical database transactions: Identification of probable frauds using machine learning for user profiling. In *Proceedings of the 1997 IEEE Knowledge and Data Engineering Exchange Workshop*, KDEX '97, pages 107–, Washington, DC, USA, 1997. IEEE Computer Society.

68. M. Kubat, R.C. Holte, S. Matwin, R. Kohavi, and F. Provost. Machine learning for the detection of oil spills in satellite radar images. *Machine Learning*, 30(2):192–215, 1998.
69. Daniel T. Larose. *Discovering Knowledge in Data: An Introduction to Data Mining*. Wiley-Interscience, 2004.
70. Wenke Lee and Dong Xiang. Information-theoretic measures for anomaly detection. In *In Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pages 130–143, 2001.
71. W. Li, J. Han, and J. Pei. CMAR: Accurate and efficient classification based on multiple class-association rules. In *Proc. of IEEE Int. Conf. on Data Mining*, pages 369–376, 2001.
72. Tjen-Sien Lim, WEI-YIN LOH, and W. Cohen. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms, 2000.
73. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *Proc. of ACM SIGKDD Int. Conf. on Kwnoledge Discovery and Data Mining*, pages 80–86, 1998.
74. B. Liu, Y. Ma, and C.K. Wong. Improving an association rule based classifier. In *Proc. of Principles of Data Mining and Knowledge Discovery*, pages 504–509, 2000.
75. Artis M. Ayuso M. & Guillen M. Modelling different types of automobile insurance fraud behaviour in the spanish market., 1999.
76. Maloof M. Learning when data sets are imbalanced and when costs are unequal and unknown, 2003.
77. Sam Maes, Karl Tuyls, Bram Vanschoenwinkel, and Bernard Manderick. Credit card fraud detection using bayesian and neural networks. In *In: Maciunas RJ, editor. Interactive image-guided neurosurgery. American Association Neurological Surgeons*, pages 261–270, 1993.
78. Jesus Mena. *Investigative Data Mining for Security and Criminal Detection*. Butterworth-Heinemann, Newton, MA, USA, 2002.
79. Lindsay C.J. Mercer. Fraud detection via regression analysis. *Comput. Secur.*, 9:331–338, May 1990.
80. S. Muggleton. Inductive Logic Programming: Theory and methods. *The Journal of Logic Programming*, 19-20:629–679, May 1994.
81. Uzi Murad and Gadi Pinkas. Unsupervised profiling for identifying superimposed fraud. In *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '99, pages 251–261, London, UK, 1999. Springer-Verlag.
82. Chawla N. C4.5 and imbalanced data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure, 2003.
83. Levine N. Crimestat: A spatial statistics program for the analysis of crime incident locations, 1999.
84. NetMap. Fraud and crime example brochure., 2004.
85. Thomas Ormerod, Nicola Morley, Linden Ball, Charles Langley, and Clive Spenser. Using ethnography to design a mass detection tool (mdt) for the early discovery of insurance fraud. In *CHI '03 extended abstracts on Human factors in computing systems*, CHI '03, pages 650–651, New York, NY, USA, 2003. ACM.

86. Claudia Perlich and Foster Provost. Aggregation-based feature invention and relational concept classes. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '03, pages 167–176, New York, NY, USA, 2003. ACM.
87. Bernhard Pfahringer. A new mdl measure for robust rule induction (extended abstract). In *Proc. of European Conf. on Machine Learning*, pages 331–334, 1995.
88. Clifton Phua, Damminda Alahakoon, and Vincent C. S. Lee. Minority report in fraud detection: classification of skewed data. *SIGKDD Explorations*, 6(1):50–59, 2004.
89. D. Potts. Incremental learning of linear model trees. In *Proc. of Int. Conf. on Machine learning*, pages 84–91, 2004.
90. Foster Provost and Tom Fawcett. Robust Classification for Imprecise Environments. *Machine Learning*, 42(3):203–231, March 2001.
91. Foster Provost and Venkateswarlu Kolluri. A survey of methods for scaling up inductive algorithms. *Data Min. Knowl. Discov.*, 3:131–169, June 1999.
92. J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5(3):239–266, 1990.
93. J. R. Quinlan. Learning with continuous classes. In *Proc. of Australian Joint Conf. on Artificial Intelligence*, pages 343–348, 1992.
94. J. Ross Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann, 1993.
95. Brockett P. Xia X. & Derrig R. Using kohonen's self organising feature map to uncover automobile bodily injury claims fraud., 1998.
96. P. Riddle, R. Segal, and O. Etzioni. Representation design and brute-force induction in a boeing manufacturing domain. *Applied Artificial Intelligence*, 8(1):125–147, 1994.
97. Saharon Rosset, Uzi Murad, Einat Neumann, Yizhak Idan, and Gadi Pinkas. Discovery of fraud rules for telecommunications-challenges and solutions. In *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 409–413. ACM Press, 1999.
98. He H. Wang J. Graco W. & Hawkins S. Application of neural networks to detection of medical fraud., 1997.
99. Pathak J. Vidyarthi N. & Summers S. A fuzzy-based algorithm for auditors to detect element of fraud in settled insurance claims, 2003.
100. Hua Shao, Hong Zhao, and Gui R. Chang. Applying data mining to detect fraud behavior in customs declaration. In *Proceedings of the 2002 International Conference on Machine Learning and Cybernetics*, volume 3, pages 1241–1244, November 2002.
101. Burge Shawe-Taylor, P. Burge, J. Shawe-taylor, Y. Moreau, H. Verrelst, C. Stoermann, and P. Gosset. Brutus - a hybrid detection tool., 1997.
102. E. Sherman. Fighting web fraud., 2002.
103. P. Smyth and R. M. Goodman. An information theoretic approach to rule induction from databases. *IEEE Transactions on Knowledge and Data Engineering*, 4(4):301–316, 1992.
104. SPSS. www.spss.com/clementine.
105. Deshmukh A. & Talluru T. A rule based fuzzy reasoning system for assessing the risk of management fraud, 1997.
106. Kim M. & Kim T. A neural classifier with fraud density map for effective credit card fraud detection., 2002.

107. Michiaki Taniguchi, Michael, Michael Haft, Jaakko Hollmn, and Volker Tresp. Fraud detection in communications networks using neural and probabilistic methods. In *Proceedings of the 1998 IEEE Int*, pages 1241–1244, 1998.
108. F. Thabtah. A review of associative classification mining. *The Knowledge Engineering Review*, 22(1):37–65, 2007.
109. Stijn Viaene, Richard A. Derrig, Bart Baesens, and Guido Dedene. A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection, 2002.
110. Stijn Viaene, Richard A. Derrig, and Guido Dedene. A case study of applying boosting naive bayes to claim fraud diagnosis. *IEEE Trans. on Knowl. and Data Eng.*, 16:612–620, May 2004.
111. Constantin von Altrock. *Fuzzy logic and NeuroFuzzy applications in business and finance*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1997.
112. Haixun Wang, Wei Fan, Philip S. Yu, and Jiawei Han. Mining concept-drifting data streams using ensemble classifiers. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235, New York, NY, USA, 2003. ACM.
113. Y. Wang and I. H. Witten. Inducing model trees for continuous classes. In *Proc. of European Conf. on Machine Learning*, pages 128–137, 1997.
114. Takashi Washio and Hiroshi Motoda. State of the art of graph-based data mining. *SIGKDD Explor. Newsl.*, 5(1):59–68, July 2003.
115. G. Webb, J. Boughton, and Z. Wang. Not so naive bayes: Aggregating one-dependence estimators. *Machine Learning*, 58(1):5–24, 2005.
116. Gary Weiss. Mining with Rarity: A Unifying Framework.
117. G.M. Weiss. Mining with rarity: a unifying framework. *SIGKDD Explorations*, 6(1):7–19, 2004.
118. G.M. Weiss and H. Hirsh. A quantitative study of small disjuncts. In *Procs AAAI*, pages 665–670, 2000.
119. G.M. Weiss and F. Provost. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 19:315–354, 2003.
120. Weka Machine Learning Project. Weka. URL http://www.cs.waikato.ac.nz/~ml/weka.
121. Richard Wheeler, Stuart Aitken, and South Bridge. Multiple algorithms for fraud detection. *Knowledge-Based Systems*, 2000:93–99, 2000.
122. Graham J. Williams. Evolutionary hot spots data mining - an architecture for exploring for interesting discoveries. In *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, PAKDD '99, pages 184–193, London, UK, 1999. Springer-Verlag.
123. Ian H. Witten and Eibe Frank. *Data Mining: Practical Machine Learning Tools*. Morgan Kaufmann, 2005.
124. D. H. Wolpert and W. G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 1(1):67–82, April 1997.
125. X. Xin and J. Han. CPAR: Classification based on predictive association rules. In *Proc. of SIAM Int. Conf. on Data Mining*, pages 331–335, 2003.
126. Syeda M. Zhang Y. & Pan Y. Parallel granular neural networks for fast credit card fraud detection., 2002.
127. Kenji Yamanishi, Jun-Ichi Takeuchi, Graham Williams, and Peter Milne. On-line unsupervised outlier detection using finite mixtures with discounting learning algorithms. *Data Min. Knowl. Discov.*, 8:275–300, May 2004.

128. Bianca Zadrozny and Charles Elkan. Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 204–213, New York, NY, USA, 2001. ACM.