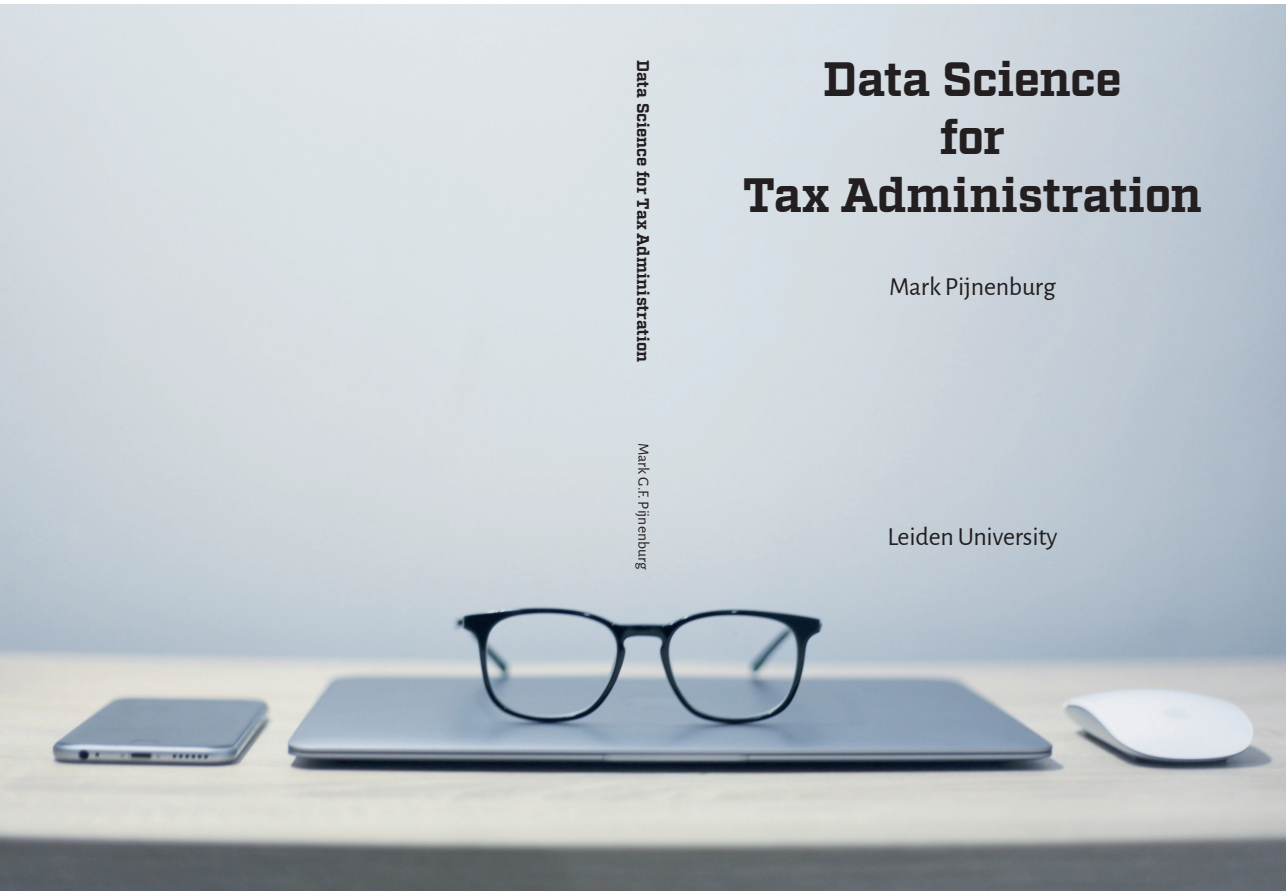
Data science for tax administration

Pijnenburg, M.G.F.



Data Science for Tax Administration

Mark C.J.F. Pijnenburg

Data Science for Tax Administration

Mark Pijnenburg

Leiden University

Data Science for Tax Administration

Mark Gerardus Franciscus Pijnenburg

Universiteit
Leiden

Cover background image: Black framed eyeglasses by Jesus Kiteque on Unsplash
Typeset using LaTeX
Printed by Ridderprint — www.ridderprint.nl
ISBN 978-94-6375-942-7

Data Science for Tax Administration

Proefschrift

ter verkrijging van
de graad van Doctor aan de Universiteit Leiden,
op gezag van Rector Magnificus prof.mr. C.J.J.M. Stolker,
volgens besluit van het College voor Promoties
te verdedigen op woensdag 24 juni 2020
klokke 15:00 uur

door

MARK GERARDUS FRANCISCUS PIJNENBURG

geboren te Heerlen
in 1979

Promotores:	Prof. dr. ir. W. Kraaij
	Prof. dr. J.N. Kok
Co-promotor:	Dr. W.J. Kowalczyk

Promotiecommissie:
Prof. dr. A. Plaat (voorzitter)
Prof. dr. T.H.W. Bäck (secretaris)
Prof. mr. dr. E.C.J.M. van der Hel - van Dijk RA (Nyenrode Business Universiteit)
Prof. dr. H. Blockeel (KU Leuven)
Prof. dr. A.P.J.M. Siebes (Universiteit Utrecht)
Prof. dr. ir. R. Arendsen
Dr. C.J. Veenman
Dr. A.J. Knobbe

Contents

1 Introduction

1.1 The Historical Link between Taxes and Data

Taxes and data have a close bond for centuries. The linking pin is called 'administration', see the relation below.

$$\text{taxes} \longleftarrow \text{administration} \longleftarrow \text{data}$$

To levy taxes an administration is required, for instance an administration of properties, transactions, or income. These administrations are basically just collections of data.

The link between taxes and data has emerged already at the start of civilization. Figure 1.1 displays an example of a Mesopotamian clay tablet. These tablets show commodities, like jars, cattle, and grain. Scholars now know that these tablets were mainly used for tax and administration purposes [12].

Also in the Roman times, the link between taxes and data is present. Most readers will recall the story about the birth of Jesus Christ, see Figure 1.2. At the time of Jesus' birth, Joseph and Maria had to travel from Nazareth to Bethlehem. Why? A census had been demanded by the Romans. And why did the Romans do all the efforts to organize a census in their great empire? Because of tax purposes. Jesus was born in the time that Augustus was the first emperor of Rome (reigning from 27 BC until his death in AD 14). Augustus changed the tax system such that each province was required to pay a flat poll tax on each adult [106]. For this reason it was important to know the exact number of adults. So also in the Roman times, one of the best examples of data collection, a census, is closely related to taxation.

In later times, data *analysis* emerged. If data has been gathered consistently for some time, it gives us the possibility to analyze the data and try to see patterns and

Figure 1.1: Clay tablet from Uruk in cuneiform writing describing commodities *source: Metropolitan Museum of Art, accession number 1988.433.3*.

learn things about the world around us. Much of human knowledge has been collected in this way.

Governments are among the early adopters of analyzing data to gain insights. An early example of statistical inference is the Trial of the Pyx [112], an annual procedure that is held in the United Kingdom since 1282 AD, see Figure 1.3. The procedure is held by Her Majesty's Treasury, the governmental department whose minister is also responsible for the British tax authority. The goal of the procedure, that still exists today, is to ensure that newly minted coins conform to the required standards, in particular whether the required amount of precious metals are used. The procedure is among the oldest statistical sampling methods.

Another example of the early involvement of governments in data analysis can still be seen in the origins of the word 'Statistics'. The word derives ultimately from the Latin *statisticum collegium* ('council of state') [81]. The German word 'Statistik', first introduced by Gottfried Achenwall in the mid eighteenth century, originally designated the analysis of data about the state. It acquired the meaning of the collection and analysis of data in the early 19th century [111]. According to the online encyclopedia Wikipedia, *Thus, the original principal purpose of Statistik was data to be used by governmental and (often centralized) administrative bodies*.

The link between taxes, administration, and data has stood the test of time as tax administrations are today still large, data processing organizations. Moreover it is inspiring to see that also today, governments, and in particular tax administrations, are heavily interested in the fruits that data science can bring, see section 1.3 for the motivation of the Netherlands Tax and Customs Authority (NTCA) [79] for doing so.

Figure 1.2: Maria and Joseph on their way to Bethlehem. *Unknown Artist*

1.2 Tax Administrations in the Computer Age

A major task of tax administrations is to make it possible for the people to pay taxes and to check whether they do. To fulfill this task, many processes have to be organized, like receiving and processing tax returns. The processes themselves generate data (incoming tax returns for example) or need data (banking data for instance). It is thus logical that an organization that processes so much data can be helped enormously by computer power.

The arrival of computers, at the end of the twentieth century, has revolutionized data processing, also in the field of public administration and taxes. Computers allow for the storage of massive data sets and have replaced extensive paper archives. The new computing power has replaced the manual calculations of taxes and workflow management software has digitized processes within modern tax administrations.

The early applications of computers in tax administrations have been to store data and automate transaction processes. However, in the last decade tax administrations started to realize that data may provide valuable information about the efficiency of processes and interesting insights about the behavior and needs of taxpayers. Information that can be used for instance to cut inefficiencies or increase services.

Tax administrations are in fact natural places to exploit the benefits of data analysis. Tax administrations have access to a lot of relevant data. To give an impression, the Netherlands Tax and Customs Administration has access to, among others,

- Information about incomes,
- Financial information about companies like profits, costs, and turnover,

Figure 1.3: Trial of the Pix. By Matt Brown - https://www.flickr.com/photos/londonmatt/3269860504, CC BY 2.0, https://commons.wikimedia.org/w/index.php?curid=52049921.

- Real estate information,
- Municipal data, like addresses, and (family) relations,
- Chamber of Commerce data,
- Counter-information from abroad,
- Some bank details.

See the information brochure [13] for a complete overview. Of course, this data can only be used for data analysis if there exists a legal basis.

Tax administration are also suitable places for data analysis, since a tax administration can make the required investments. Moreover, a tax authority may benefit from even a small increment in efficiency due to the vast sums of money that are processed. Besides, most tax administrations have a large, in-house ICT department, facilitating access to the required technical facilities. Tax administrations also fit with the required fact-driven culture, stemming in part from the nature of the business and the presence of many accountants.

1.3 Analytics at the Netherlands Tax and Customs Administration

The Netherlands Tax and Customs Administration (NTCA, or 'Belastingdienst' in Dutch) is the Dutch governmental organization responsible for collecting the main taxes in the Netherlands [79]. The organization falls under the responsibility of the Ministry of Finance. Its main tasks are to make it possible for taxpayers to pay taxes and to check whether this is done. Customs has been part of the NTCA since its founding in 1806 by finance minister Alexander Gogel, see Figure 1.4.

Figure 1.4: Portrait of Alexander Gogel, Minister of Finance, who founded the Netherlands Tax and Customs Administration in 1806. Source: Belasting & Douane Museum

Today, the organization collects about €200 billion annually and employs slightly less than 30,000 employees. It not only collects taxes but also hands out benefits. Benefits are income-dependent allowances from the government in, for example, healthcare costs, childcare costs and housing costs (rent allowance). The NTCA is present in most cities in the Netherlands, see Figure 1.5.

At least two major trends have contributed to the acceleration of analytics at the NTCA, that has started at about 2013. The first trend has its origin in the financial crisis of 2008. At that moment the Dutch government took a number of austerity measures, including major budget cuts for the various departments. These budget cuts were spread out over a five-year period, leading to ever severe budget cuts each year, increasing the need for efficient operations. The second major trend has to do

Figure 1.5: Locations of the NTCA in four major cities: Amsterdam (top left), Groningen (top right), Apeldoorn (bottom left), and Utrecht (bottom right).

with the age structure of the Tax and Customs Administration staff, which means that since 2016 around 2000 employees retire yearly. This urged the top management to invest in promising techniques that could increase effectiveness.

Another long term trend has been the steady increase of the workload for the NTCA without a steady increase in the number of employees, see Figure 1.6. The figure shows that the number of entrepreneurs have tripled in the 27 year period from 1986 to 2013, a consequence of economic growth but also of a policy of stimulating entrepreneurship by the government. Many of these entrepreneurs are self-employed and take more effort to check for the tax administration compared to employees. The figure also highlights the doubling of the number of income tax returns in this period.

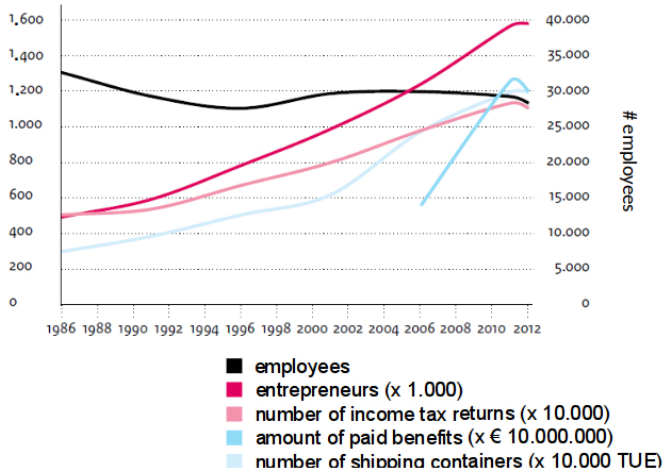


Figure 1.6: Trends related to the workload of the Netherlands Tax and Customs Administration over a 27-year period (1988 - 2013). The colored lines (blue and pink) depict trends that have a major influence on the workload. The scale belonging to the colored lines is depicted on the left vertical axis. The black line represents the number of Full Time Equivalent (FTE) employees. Its scale is depicted on the right vertical axis.

This rise has partly to do with a rise in the general population, but also with the fact that the lifestyle of citizens has become more dynamic and thus more citizens are asked to send in a tax return with detailed information. Also visible in the figure is the fact that since 2006 the NTCA was asked to take over the payment of several benefits from other departments. Finally the figure shows that the number of containers entering the Port of Rotterdam has more than tripled in this period, an indication of growing internationalism, but also of a lot more work for Customs. The black line in Figure 1.6 represents the number of employees in this time period, showing a constant number near 30.000. Considering these numbers, the interest and support of the NTCA for analytics is clear.

1.4 Analytics and Related Concepts

Analyzing data in the computer age comes down to extracting information from the raw data stored in computer systems. The systematic retrieval of data and the transformation of data into information and relevant knowledge is called *data mining*. Although various tax administrations have reached different levels of maturity in data

mining, it is safe to state that currently, no tax administration has fully exploited the information embedded in the data.

Extracting insights from data does not yet create tangible business value. For instance, insights should be 'actionable', i.e. management must be able to change current practices based on the insights. To maximally reap the (business) benefits of data mining, many activities have to be organized beyond the data mining itself, and strategic choices have to be made. Davenport and others [32, 33, 31] have studied data mining in organizations from a business science perspective. They call the application of techniques from statistics and machine learning on data in order to improve business processes *analytics*.

Recently, the vocabulary around data analysis has expanding further with terms as *deep learning*, *big data*, *machine learning*, *advanced analytics*, and *predictive modelling*. Figure 1.7 is an attempt to explain the term 'Analytics' in relation to 'Statistics', 'Machine Learning' and 'Data Science' as an understanding of these relations helps to avoid confusion while reading this thesis.

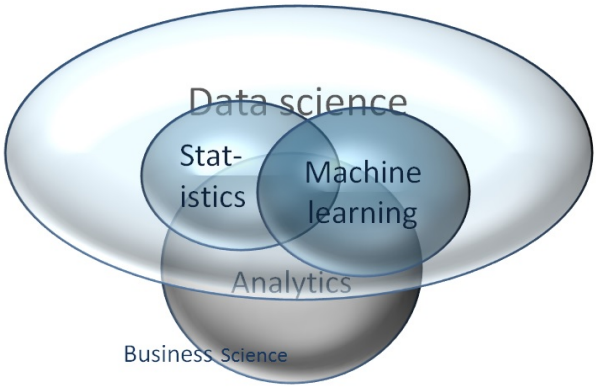


Figure 1.7: Venn-diagram visualization of the concepts *Analytics*, *Data Science*, and *Machine Learning*

The core of the diagram is formed by the components 'Statistics' and 'Machine learning'. These are fields of research that have produced a large collection of techniques that are helpful in analyzing data. Statistics is the oldest discipline of the two and has strong connections with mathematics, especially probability theory. Statistics has introduced many important basic concepts in data analysis. It originated in the pre-computer era, and presumably therefore its foundations and methods are more inclined to rigorous, analytical (i.e. closed form) solutions over heuristic algorithms that come to solutions by exploiting computing power. The connections with mathematics makes that statisticians are likely to think in probabilistic models underlying

a data set.

Machine Learning is a younger scientific discipline and has close connections with Computer Science and (Electrical) Engineering. Machine learning often puts emphasis on challenging practical problems that mimic human intelligence, like image recognition, natural language processing, automatic translation, speech recognition and learning strategies to play games. The connection with computer science makes machine learning in general more 'computation-intensive'. The influence of engineering makes the discipline practical, preferring in general working solutions over theoretical achievements. This said, the boundaries between the two disciplines in modern research are often hard to distinguish.

Around the core in Figure 1.7 the large component called 'Data Science' is shown. Data Science is a broad concept that roughly contains all activities that have to do with the transformation of data to knowledge. It is broader than statistics and machine learning as it does also include techniques to extract data and transform data. In that sense it resembles Data Mining. However it also covers applications of the findings of data analysis and as such has an overlap with many application domains, like geology, medicine, biology, social sciences, and engineering. In fact, in a funny, extreme sense, one could argue that all scientific disciplines are practising data science as long as they infer conclusions from data.

The last component in Figure 1.7, Analytics, was mentioned above. Analytics is understood to be the application of techniques from statistics and machine learning to add value to organizations. Analytics pays attention to embed knowledge from data in existing or new business processes. Analytics states that organizations can achieve a strategic advantage by excelling in data analysis.

1.5 Research Questions

In this section we position the topics of this thesis in the wider field of scientific research. This way it becomes clear where our results fit within the vast body of knowledge already present. To achieve this, we first shortly sketch the research disciplines of the administration of taxes as well as data science. Subsequently, we will formulate the research problem and the related research questions. Finally we will fit the research questions into the framework provided by the Organization of Economic Co-operation and Development (OECD).

Taxation is a large and flourishing research field as is evident from a long list of specialized journals like *Fiscal Studies*, *The National Tax Journal*, *Tax Law Review* and many, many more. Also the more practical side of taxation, namely the administration of taxes, receives considerable attention, see for instance the many articles on the

subject in the *Journal of Public Economics* and the *Journal of Tax Administration*. A scientific conference on the administration of taxes exists as well: the *International Conference on Tax Administration*.

Data Science is a flourishing research field as well. In the field of machine learning (a major sub-discipline of data science, see Figure 1.7), peer-reviewed conferences play a fundamental role. Some well-known, large conferences are the *International Conference on Machine Learning*, the *European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases* and the *Conference on Neural Information Processing Systems*. Some well-known journals in the field are the *Journal of Machine Learning Research*, *Machine Learning*, *Statistical Modelling*, and others.

At the intersection of the two academic disciplines of the 'administration of taxes' and 'data science', surprisingly little research is available. Currently, there is no scientific journal or conference specialized in this topic. A reason may be that tax data is typically not made available publicly, which makes publication hard as scientific results need to be reproducible. Although some journals publish articles on the application of data science to tax administration (notably *Expert Systems with Applications*, and *Electronic Journal on e-Government* as well as some conferences in machine learning), the application area of data science to tax administration is currently small.

This leads us to the research problem that underlies this thesis:

> **Research problem** *Expand our knowledge of the applications of data science to improve the administration of taxes in terms of effectiveness, efficiency, or improved compliance.*

The research problem is of interest as the potential of data science to the administration of taxes looks promising (see sections 1.2, 1.3, as well as the general news items on the potential of data science for data-rich domains), while many tax administrations are still at the start or intermediate level of implementing data science techniques [40] in their day-to-day business.

The research problem is too broad to answer in a single PhD-track. For this reason we limited the scope of this thesis to the research questions below that are all directly related to the research problem:

Research Questions

1. What data science / analytical techniques can be used in *taxpayer supervision* and what contributions may be expected from these techniques (Chapter 2)?
2. Can we improve on current audit selection models by incorporating categorical variables with many values (like 'industry sector') that are valued by experts doing manual audit selection (Chapter 3)?
3. Can we develop a new anomaly detection algorithm tailored to find the outliers that are of interest to a tax administration (Chapter 4)?
4. Unsupervised anomaly detection algorithms play an important role in (tax) fraud detection. What algorithms can be expected to work well under what conditions (Chapter 6)?
5. Can we develop a data science technique that assists in testing similar treatment of similar (tax) cases (Chapter 5)?

During the PhD track we have come across some smaller topics that are worth noting, but are not directly related to the research questions formulated above. These topics will be covered in Chapter 7.

A framework for the research at the intersection of data science and the administration of taxes, can be found in the OECD report *Advanced Analytics for Better Tax Administration; Putting Data to Work* [40]. The OECD is an established intergovernmental organization that facilitates cooperation between 36 industrialized countries. Its objective is to shape policies that foster prosperity, equality, opportunity and well-being for all. This objective is achieved, among others, by setting standards and exploring new developments.

According to Chapter 2 of the OECD-report *Advanced Analytics for Better Tax Administration*, analytics can be applied within the administration of taxes to the following tasks:

a audit case selection,
b filing and payment compliance,
c debt management,
d taxpayer services,
e policy evaluation,
f taxpayer segmentation.

Some tasks above have a clear link to effectiveness ('audit case selection', 'taxpayer segmentation', 'taxpayer segmentation'), while others are more focused on improving taxpayer compliance ('filing and payment compliance', 'taxpayer services') or efficiency ('debt management').

When considering these tasks in the light of the general distinction of analytical tasks between internal and external processes (more details can be found in Chapter

Task	Links to	Research Question
a. audit case selection	effectiveness	2
b. filing and payment compliance	compliance	5
c. debt management	efficiency	none, but small contribution in section 7.2
d. taxpayer services	compliance	none
e. policy evaluation	effectiveness	1
f. taxpayer segmentation	effectiveness	none, but small contribution in section 7.4

Table 1.1: Relation of research question to OECD tax administration tasks suitable for data science applications

2.1), made by Davenport and Harris [32], it might be said that items 'b' and 'e' belong mainly to internal processes, while items 'a','c', 'd', and 'f' are more directed to external processes of a tax administration.

The relation between the research questions and the six main tasks of tax administrations above, is shown in Table 1.1. The table reveals that most tasks have been touched in this thesis. It also becomes clear that research questions 3 and 4, that deal with fraud detection, do not fit in the table. One could argue that fraud detection is part of audit case selection, or one could add fraud detection as an additional task of tax administrations suitable for improvement by data science.

1.6 Outline of Thesis

As explained in the previous section, four main topics have been studied:

- The role of analytics in taxpayer supervision (Chapter 2),
- Categorical features with many levels in risk models (Chapter 3),
- Anomaly detection (Chapters 4 and 6), with possible applications for fraud detection or data quality,
- Process mining to test the similar treatment of similar cases (Chapter 5).

The first topic, the role of analytics in taxpayer supervision, is at the intersection of computer science and business administration. The chapter addresses the first research question. The interest in the subject stems from two opposite opinions with respect to data science and tax administrations that could be heard some years ago at different tax administrations. One opinion put analytics away as 'another hype'. The

other put analytics on a pedestal and expected great results. By looking more closely at analytics techniques and comparing these with the activities associated with taxpayer supervision, a clearer picture has been created of what one can realistically expect from analytics.

The second topic combines the recently developed technique of Factorization Machines with the classical classification algorithm of logistic regression and thereby answers the second research question. The combination allows logistic regression models to incorporate interactions of categorical features with many levels, without the need for an excessive amount of data. When developing risk models at the NTCA, these categorical variables appeared, but could not be handled with classical techniques.

The third topic contains two contributions to the field of anomaly detection. The first contribution, in Chapter 4, answers research question 3. In this chapter a particular class of anomalies, called *singular anomalies*, are put forward. This type of anomalies are typical encountered when dealing with tax evasion. These anomalies are observations characterized by a single or a few anomalous features, while the other features display regular values. An algorithm is presented for finding these anomalies. The second contribution, in Chapter 6, adds knowledge to the anomaly detection domain by testing three anomaly detection algorithms on the recently published benchmark of Goldstein and Uchida [43]. This chapter answers research question number 4.

The fourth topic (see Chapter 5) answers the last research question. The topic is in the field of process mining, a relatively young branch of data science that takes process logs as input and tries to describe and optimize (business) processes. We contributed a test procedure to verify, based on event logs, whether two groups of process workers treat similar cases similarly. Treating similar cases in a similar way is part of the permanent task of the NTCA.

Besides the four main topics mentioned above, we have touched some others as well. The contributions to these topics are limited, so none of them deserves a chapter of its own. Nevertheless we felt that we had to include these topics, mainly to show the reader the broadness of the field of applications of data science to tax administration. We have collected these topics in a final chapter, Chapter 7. These side-topics are:

- application of analytics within Human Resources,
- reinforcement learning applied to tax debt collection,
- explaining risk models to non-experts, and
- recommendations based on fuzzy sets.

1.7 Publications Related to Thesis

The content of this thesis is based on the following, previously published articles (in chronological order).

- C. Boon, F. D. Belschak, D. N. Den Hartog, and M. Pijnenburg. Perceived human resource management practices. *Journal of Personnel Psychology*, 2014
- M. Pijnenburg and W. Kowalczyk. Applying analytics for improved taxpayer supervision. In *Proceedings of 16th European Conference on e-Government ECEG 2016*, pages 145–153. Academic Conferences and publishing limited, 2016
- M. Pijnenburg, W. Kowalczyk, E. van der Hel-van Dijk, et al. A roadmap for analytics in taxpayer supervision. *Electronic Journal of e-Government*, 15:19–32, 2017
- M. Pijnenburg and W. Kowalczyk. Extending logistic regression models with factorization machines. In *International Symposium on Methodologies for Intelligent Systems*, pages 323–332. Springer, 2017
- M. Pijnenburg and W. Kowalczyk. Singular outliers: Finding common observations with an uncommon feature. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 492–503. Springer, 2018
- M. Pijnenburg and W. Kowalczyk. Are similar cases treated similarly? a comparison between process workers. In *International Conference on Business Information Systems*, pages 1–15. Springer, 2019
- M. Pijnenburg and W. Kowalczyk. Extending an anomaly detection benchmark with auto-encoders, isolation forests, and rbms. In *International Conference on Information and Software Technologies*. Springer, 2019. Best Paper Award

The side-topic of applying reinforcement learning to the collection of taxes has led to the master thesis 'Tax data and reinforcement learning', written by Martijn Post under the author's supervision. The side-topic of explaining risk models to non-experts has led to a non-scientific publication,

- M. Pijnenburg and K. Kuijpers. Explaining risk models to the business. *Tax Tribune*, 35:57–62, 2016.

A full list of publications of the author can be found at page 141 of this thesis.

2 Analytics in Taxpayer Supervision

In this chapter opportunities for applying analytics within taxpayer supervision (also known as 'compliance risk management for tax administrations') are explored. The research in this chapter is guided by the research question:

> **Research Question** *What data science / analytical techniques can be used in* taxpayer supervision *and what contributions may be expected from these techniques?*

The general idea of applying analytics is made more concrete for taxpayer supervision by explicitly writing down the tasks of taxpayer supervision and the techniques known from analytics and data science. This will lead to more insight into what we may expect from analytics and will assist tax administrations that want to improve their analytical capabilities. Also, an overview is given of the current state of analytics in tax administrations. Attention is paid as well to the limitations of analytics. Findings include that over half of the activities in taxpayer supervision can be supported by analytics. Additionally, a match is presented between supervision activities and specific analytical techniques that can be applied for these activities. The chapter also presents a short case study of the Netherlands Tax and Customs Administration on the selection of VAT refunds. The chapter is based on the following articles:

- M. Pijnenburg and W. Kowalczyk. Applying analytics for improved taxpayer supervision. In *Proceedings of 16th European Conference on e-Government ECEG 2016*, pages 145–153. Academic Conferences and publishing limited, 2016
- M. Pijnenburg, W. Kowalczyk, E. van der Hel-van Dijk, et al. A roadmap for analytics in taxpayer supervision. *Electronic Journal of e-Government*, 15:19–32, 2017

2.1 Introduction

2.1.1 Research objective

In this chapter we investigate what analytical / data science techniques can be used in taxpayer supervision and what contribution may be expected from these techniques. Several theories exist on how to effectively supervise taxpayers, see Section 2.2.1. In this chapter we will focus on *compliance risk management*, a modern theory about taxpayer supervision that is adopted by many western tax authorities.

The investigation will give directions to tax administrations willing to improve their analytical capabilities in taxpayer supervision. It also offers some insights to researchers in e-Government with interest in the potential of analytics for governmental organizations.

To reach the research objective, the terms 'analytics' and 'taxpayer supervision' are decomposed into underlying techniques and activities, based on the available literature. Subsequently, the techniques are mapped to supervision activities, according to their relevance and suitability. To illustrate the practical side of analytics, a short case study is included.

2.1.2 Need for more effective taxpayer supervision

Taxpayer supervision needs to become more effective due to an expanding workload often combined with staff reduction and budget cuts. Workload increases by a growing number of taxpayers – both private individuals and businesses – and a rise in dynamics of the taxpayer population (e.g. shifting from employment to self-employed and vice versa). Moreover, the workload of taxpayer supervision expands by growing international trade, partly due to new developments in e-commerce [62, 41]. Another reason to improve the effectiveness of taxpayer supervision is the rising expectations of citizens that want cheap, high-quality government agencies. Rising expectations of citizens are partly due to higher education levels [41] combined with the experience of smoothly operating non-governmental organizations and businesses.

'Analytics' is a promising candidate for improving the effectiveness in taxpayer supervision. Davenport and Harris [32] define 'analytics' as *extensive use of data, statistical and quantitative analysis, explanatory and predictive models, and fact-based management to drive decisions and actions*, and we will follow this definition in this chapter. Decisions and actions that result from an analytical approach have often led to more effective processes [32] in organizations, that are similar to tax administrations concerning their size and activities. Moreover, tax administrations meet an essential condition for starting with analytics, namely the availability of data: tax administrations

generate many transaction data and have access to much third-party data. As a side effect, analytics may increase objectivity of the treatment of taxpayers.

The interest of tax administrations in 'analytics' or 'data exploitation' is therefore evident. International bodies like the Organisation for Economic Co-operation and Development (OECD), the European Commission (EC), and the Intra-European Organisation of Tax Administrations (IOTA) have put analytics on their agenda. Moreover, several tax administrations (among others the tax administrations of The Netherlands and the United Kingdom) are investing considerably to reap the benefits of analytics.

2.1.3 Scope of the research

In this chapter, applications of analytics are restricted to taxpayer supervision. We may position taxpayer supervision among other activities of a tax administration by considering the following dichotomy. In general, a tax administration has the following two tasks: (1) to make it possible for taxpayers to pay taxes, and (2) to examine whether taxpayers paid them. The first task requires a proper organization of internal processes like a tax return filing process and a payment process. The second task requires an adequate supervision process. These two main tasks of a tax administration coincide largely with a distinction made by Davenport and Harris [32]. They distinguish between applications of analytics to improve internal processes (financial, manufacturing, Research & Development, and Human Resources) and external processes (customer and supplier processes). As taxpayer supervision is an external process, we will leave out internal processes in this chapter.

Note that the term 'taxpayer' is used broadly in this chapter, as a term for a private individual, a business, a corporation, or any other legal entity that is taxable.

2.1.4 Organization of this chapter

The chapter is organized as follows. Section 2.2 explores related research. Section 2.3 sketches current developments in analytics in tax administrations and the insurance and banking sector. Section 2.4 describes everyday activities in taxpayer supervision and regular classes of analytical techniques. The techniques are subsequently mapped onto the activities and a roadmap for analytics in taxpayer supervision is sketched. Section 2.5 presents a case study of the Netherlands Tax and Customs Administration (NTCA) aimed at detecting erroneous VAT refunds [14]. Section 2.6 contains conclusions, and Section 2.7 provides a discussion on analytics in taxpayer supervision.

2.2 Related Research

2.2.1 Theories about modern supervision

There exists a rich literature about tax compliance, starting with the seminal paper of Allingham and Sandmo (1972) [5] in which the authors discuss the economics-of-crime theory. This theory looks at a taxpayer as a 'homo economicus', deliberately weighing the expected utility before deciding to comply with the tax laws (or not). Therefore, tax administrations use so-called 'deterrence' strategies, based upon the assumption that the threat of detection and punishment enforces compliance. In this view, the frequency of audits and the size of fines are tools for treating non-compliance. Analytics might contribute to such a strategy, by optimizing the selection of taxpayers for audits, detecting fraud, and calculating optimal values of fines.

However, in practice, observed compliance levels proved to be higher than predicted by this early theory. This empirical fact gave rise to new theories about influencing tax compliance behavior. These new theories have identified many factors that play a role in the actual behavior of taxpayers (Andeoni, Erard and Feinstein, 1998 [7]), such as psychological factors, personal norms, social norms, tax morale, and opportunities for tax evasion. A review paper of Jackson and Milliron (1986) [55] summarizes tax compliance research in the period 1970 - 1985, while a review paper of Richardson and Sawyer (2001) [99] extends this period towards 2001. Alm (2012) [6] gives a more recent overview. Research showed that 'deterrence strategies' alone are unable to efficiently attain or maintain desired compliance levels (especially given a finite level of resources).

New insights in behavior generated new ideas about 'advice and persuade' strategies. Several scholars from Public Administration have suggested policies for adequate supervision, such as the theory of responsive regulation of Braithwaite (2007) [20] and the psychology of persuasion of Cialdini (2004) [27]. These policies suggest new instruments for treating non-compliance, like limiting opportunities to make errors or reducing unintentional errors by improving services. Analytics might contribute to such a strategy, for example, by providing a more accurate description of taxpayer behavior, investigating areas of frequent unintentional errors, and improving taxpayer services.

The OECD [41] and the EC [38] both encourage tax administrations to use both these strategies within a so-called Compliance Risk Management approach, in which a tax administration attunes its strategy to the taxpayer's behavior. In this paper we will discuss taxpayer supervision from the perspective of tax administrations applying a Compliance Risk Management approach.

2.2.2 Analytical applications within tax administrations

A literature search by the author at the beginning of 2017 has identified fourteen published articles in scientific journals aiming at improving the effectiveness of taxpayer supervision with analytical techniques. These articles focus on applying specific techniques. Articles treating analytics as a general concept within tax administrations were not found.

Eleven publications treat audit selection. The techniques follow developments in computer science and statistics. Publications from the 1980s treat predominantly techniques from statistics and econometrics that require limited computations. The rise of computer power in the 1990s attracted computer scientists to the topic of extracting knowledge from data. Publications on audit selection from that period onward, focus on these newer, computation-intensive techniques.

Several studies report an increasing yield of audits by using analytics in the selection process. Hsu et al. [54] report a significant increase in efficiency (63%) compared to the manual selection of audits in Minnesota (USA). Gupta and Nagadevara [45] report an increase of the 'hit rate' of up to 3.5 times compared to random audit selection of VAT returns in India. Wu et al. [113] claim an improved accuracy (i.e. less false negatives and / or less false positives) compared to a manual process in Taiwan. Da Silva, Carvalho, and Souza [30] conclude that results are auspicious for the tax administration when studying audit selection for tax refunds in Brazil.

2.2.3 Managerial literature on analytics

Analytics has received much attention in the managerial literature since the appearance of the book 'Competing on Analytics' by Davenport and Harris in 2007 [32]. In the book, Davenport and Harris point out that analytics is more than a mere collection of techniques; by adopting a strategy of incorporating these techniques consistently in decision-making processes, a competitive advantage can be created. Since then, the managerial aspect of analytics has been the subject of many articles. Many of the findings and developments on the managerial aspect, along with some concrete examples, can be found in the subsequent books of Davenport [33], [31]. Recently, review articles have been appearing, reviewing the managerial literature on analytics for sectors like Supply Chain Management [109] or E-commerce [4]. The coverage of analytics in government has been relatively weak.

2.3 Practical Experiences with Analytics

2.3.1 General experiences of tax administrations

In 2016 the OECD, Forum on Tax Administration (FTA) issued the report 'Advanced Analytics for Better Tax Administration' [40], which provides practical examples of how tax administrations are currently using advanced analytics, see also section 1.5. OECD describes 'Advanced analytics' as 'the process of applying statistical and machine-learning techniques to uncover insight from data, and ultimately to make better decisions about how to deploy resources to the best possible effect.' Especially the use of statistical techniques to make inferences about cause and effect is interesting for those tax administrations that apply a compliance risk management strategy in which they try to influence taxpayer behavior to comply with fiscal rules. The report states that advanced analytics is proving a precious tool in improving tax administration effectiveness, meaning that it allows tax administrations to achieve its goals in a better way compared to the situation not using advanced analytics. The report, however, does not make any assessment, and practical examples only limitedly support the proof of this statement.

This OECD report [40] is based upon a survey, which is completed by 16 FTA members, one of which is the Netherlands. In chapter two of the report, six areas are identified that apply analytics: audit case selection, filing and payment compliance, taxpayer's services, debt management, policy evaluation and taxpayer segmentation. According to the survey, Australia, Ireland, New Zealand, Singapore, the United Kingdom, and the United States use advanced analytics in all areas mentioned. The Netherlands uses advanced analytics in audit case selection and debt management. Almost all respondents appear to use advanced analytics to improve audit case selection. In the other areas, the use of advanced analytics seems to be less (structurally) used. Unfortunately, the survey is less specific about the extent of applying analytical activities; are we observing isolated analytical applications or is analytics fully embedded in the culture of the organization? If the latter is the case, one expects: fact-based decision making even at the strategic level, analytics that is highly integrated in the business processes, CEO passion about analytics, and broad management commitment.

The OECD report [40] concludes that in the day-to-day work, tax administrations are always making predictions and coming to conclusions about the likely impact of their activities. Advanced analytics — in the opinion of the OECD — does not aim to achieve anything fundamentally new, but it seeks to carry out these same tasks with more reliance on data and less on human judgment.

If one looks at the current situation, most tax administrations that use advanced analytics for audit case selection seem to aim to improve the identification of tax

returns or refunds/claims that might contain errors or be fraudulent. In terms of using 'predictive' analytics the current way of working does, therefore, seems to 'predict' that a tax return contains a problem, but not (yet) seems to be able to anticipate likely problems.

2.3.2 Practical experiences in banking and insurance

Banks and insurance companies are in many aspects similar to tax administrations. Banks and insurance companies are mostly large organizations, process large sums of money, have often an extensive IT department, and employ many employees with an accountancy or legal background. For this reason it is interesting to look at these sectors as well.

Analytical techniques entered the banking and insurance sectors relatively early - in the late 90's. Simple predictive models like logistic regression or decision trees were used to address marketing problems like mailing selection, cross- and up-selling, credit scoring, improving customer retention [67]. Simple cluster analysis techniques were used to partition clients into homogeneous groups. Also in this period, the first successful application of neural networks in the banking sector took place: the Hecht Nielsen Company developed a system for detecting fraud with credit card transactions [49].

Over time the usage of analytics in banking and insurance has been expanding, resulting in better management of data, more robust data analysis tools, and automation of typical analytical tasks like data pre-processing, model building, and model maintenance. However, the main areas of application of analytical techniques have not changed: marketing, fraud detection, and risk management. It is estimated that currently in the banking sector the ratio of advanced analytics to basic business intelligence, meant as analyzing historical data with data warehousing methods, is like 72% to 28% [59].

More recently, banking and insurance sectors have been applying analytics to risk-adjusted pricing, where the objective is to determine the price of a loan or an insurance policy according to the estimated risk of the individual client. This approach, due to some controversies around it, like privacy issues, is still not very popular, according to Acebedo and Durnall [2]. For example, some insurance companies offer so-called 'user-based' car insurance, where the insurance fee is determined by the driving style that is measured by dedicated devices installed in a car [66]. Insurance companies also use more and more social media like Facebook or Twitter to detect fraud by comparing the client's claims to the information (s)he made publicly available [104].

2.4 Analytics for Taxpayer Supervision

In this section, we look more closely how analytics can contribute to taxpayer supervision when tax administrations are applying a Compliance Risk Management approach. Firstly — in 2.4.1 — a brief explanation is given of various activities that tax administrations apply in taxpayer supervision when adopting Compliance Risk Management. Subsequently, the technical side of analytics is unraveled in 2.4.2 by providing an overview of modern analytical techniques. Next, in 2.4.3, activities and techniques are mapped onto each other, leading to the first findings. Finally, in 2.4.4, a roadmap for applying analytics in taxpayer supervision is sketched.

2.4.1 Activities in taxpayer supervision

Many western tax authorities have designed their taxpayer supervision according to a so-called Compliance Risk Management approach. The objective of applying Compliance Risk Management is to facilitate the management of the tax administration to make better decisions. The Compliance Risk Management process helps to identify the different steps in the decision-making process. The five major steps are [38]: risk identification, risk analysis, prioritization, treatment, and evaluation. The first step, risk identification, aims to identify specific compliance risks that a tax administration encounters. Compliance risk is here understood as a risk of a taxpayer failing to comply with the obligations of the tax law. In the second step, risk analysis, the impact of the identified risks are assessed. Moreover, the causes of the risks are examined. In the third step, prioritization, decisions are made about supervision activities that match the causes of the identified risks/taxpayer behavior. Prioritization is needed since resources for treating risks are scarce. In step four, treatment, execution of an agreed supervision strategy takes place. In step five, the effects of the treatments (and policies) are evaluated to improve future decisions.

In general, different organizational units within a tax administration perform the activities related to these five steps. Table 2.1 shows the steps and the organizational unit that could perform the related activities. If each of the five steps contains activities that can be supported by analytics — to a varying degree — a comprehensive, analytical approach to taxpayer supervision will not be restricted to one particular organizational unit within a tax administration. In Table 2.2 we will have a more detailed look at the activities in the various stages.

Table 2.2 lists the main activities for each step following the EU and OECD guides on Compliance Risk Management (EU, 2010) and (OECD, 2004a), and classifies the activities according to the value of analytics to them. The classification is based upon a) tax literature research (an article mentions the use of analytics for this activity)

Steps in Compliance Risk Management	Department involved
Risk identification	Staff
Risk Analysis	Staff
Prioritization	Management
Treatment	Operations
Evaluation	Staff

Table 2.1: Main steps in compliance risk management and typical departments involved

b) international conferences and workshops attended by the author where various tax administrations share best practices and c) desk research for similar sectors, like banking and insurance, that mention the application of certain technique in a very similar problem. The classification in the next column in 'Low', 'Medium', 'High' is based on counting the elements in two columns 'Activities supported by analytics' and 'Activities with no role for analytics'. The classification only could be done roughly because a complete overview of activities is not available and only a limited number of workshops / conferences has been attended by the author. Limitations also arise as some activities require more time than others or are considered more important than others. These two aspects have not been taken into account.

Step	Activities supported by analytics	Activities with (almost) no role for analytics	level of activities supported
Risk Identification	Horizon scans	Society support	M
	Random audits	New legislation	
	Identify new risks from data	Information from other tax administrations	
	Segmentation of taxpayers	Third-party information	
	Detecting Fraud	Signals from the shop floor	
Risk Analysis	Quantify risks with in-house or external data		H
	Hit rate scoring		
	Random audits		
	Tax gap estimations		
	Trend analysis		
	Root-cause analysis		
	Estimating costs of treatment		

Prioritization	Calculating human and other resources	Assessing political and social effects of risks	L
	Optimizing resource allocation	Developing criteria to prioritize	
		Matching causes (of risks) and instruments	
Treatment	Easy contacts	Risk transfer to other parties	L-M
	Desk audits	Changing legislation	
	Field Audits	Consultation and agreements	
	Administrating in the cloud	Fiscal education	
	Real-time checking of tax returns	Understandable legislation, tax returns and support information	
	Pre-filled tax returns	Advance ruling	
		Inventing new treatment options	
		On-site visits	
Evaluation	Outcome measurement	Plan evaluation	M
	Experimental Design of evaluation	Process evaluation	

Table 2.2: Activities in taxpayer supervision that can be supported with analytics (H = High, M = Medium, L = Low)

Looking at Table 2.2, it seems safe to state that analytics can play a role in all stages of the Compliance Risk Management approach. Analytics may support a substantial number of activities, especially in risk identification, risk analysis, and evaluation. It is also noteworthy to observe that for a substantial number of activities, analytics does not seem to have an added value (see column 'Activities with no role for analytics' in Table 2.2).

According to the experience of the author, cooperation is necessary between analysts, people from the shop floor, process experts, and experts in supervision, to maximally improve the positive effect of analytics. Analysts need to understand the data and processes by talking with domain experts to avoid severe mistakes. Moreover, experts are needed to judge the (initial) analytical results. The intense cooperation between analysts and experts is crucial in the initial, developmental stage.

2.4.2 Classes of analytical techniques

In this section, analytical techniques are grouped by the task they perform. The grouping is a result of comparing several categorizations found within textbooks covering applications of analytics (Federer, 1991 [36]; Cramer, 2003 [29]; Linoff and Berry, 2011 [67]; Larose, 2005 [61]; Liu, 2007 [69]; Leskovec, Rajaraman and Ullman, 2014 [63]). In order not to get lost in details, we have merged some classes of analytical techniques. The merging holds especially for 'descriptive statistics' and 'mining new data sources'. As a result, we distinguish the following ten major classes of analytical techniques that are seen frequently in taxpayer supervision (see Table 2.3):

Classes of analytical techniques	
1. Descriptive statistics	6. Time series analysis
2. Experimental design	7. Anomaly detection
3. Hypothesis testing	8. Recommendation systems
4. Predictive modeling	9. (social) Network analysis
5. Cluster analysis	10. Mining new data sources

Table 2.3: Overview of classes of Analytical techniques

(1) Descriptive statistics. Techniques from descriptive statistics provide fundamental insights by calculating simple summary statistics, visualizing data, or eliminating non-informative data. The latter is often called 'data reduction', or 'feature selection'. Techniques from descriptive statistics can be highly effective, despite their simplicity, and are broadly applicable. Typical techniques in this class are the construction of frequency tables or computing means and standard deviations. Also plotting histograms, bar charts and scatterplots are frequently employed. Factor analysis is a popular technique for data reduction; see [36, 29]. In taxpayer supervision, descriptive statistics are used for instance, for determining the number of offenders and the amount of lost money of a (compliance) risk.

(2) Experimental design. Surveys and experiments are often needed to gain specialized knowledge. Techniques from experimental design assist in setting up experiments that gain maximal knowledge while limiting the number of observations to be examined. Typical techniques include sampling designs and designs for controlled experiments, such as block designs, see [36]. In taxpayer supervision, experimental design can help, for instance to design random audit programs that provide more information on risks by sampling the same number of taxpayers. Another application is to design an experiment in which taxpayers are exposed to different treatments to find the most effective treatment.

(3) Hypothesis testing. Hypothesis testing is used to test whether the data supports

an assumption (for instance about the behavior of a group of taxpayers). In taxpayer supervision, this often means checking prior assumptions of experts concerning risks. Typical techniques include statistical tests like the Chi-square test (see also Chapter 5), the F-test (implicitly used in ANOVA), or some non-parametric tests. Introductory textbooks on statistics contain more information on hypothesis testing.

(4) Predictive modeling. With predictive modeling, one tries to predict a characteristic (called 'target') of a taxpayer or a tax return statement, with the help of a model. For example, in the case of tax returns, this characteristic is often defined as true or false, depending on whether the tax return contains a particular error or not. A computer algorithm automatically generates a model based on a systematic examination of historical cases with a known target. An analyst selects a suitable algorithm and sets the parameters of the algorithm. Some popular modeling techniques are decision trees, logistic regression, discriminant analysis, k-nearest neighbors, neural networks, support vector machines, and random forests. See for instance [50] for some frequently used techniques.

(5) Cluster analysis. Techniques from cluster analysis are used to group similar taxpayers or tax returns. This grouping gives more insight and allows tailored supervision approaches. Frequently used clustering techniques include K-means, BIRCH, and DBSCAN , see [29, 69].

(6) Time series analysis. Techniques from time series analysis are applied to find patterns in measurements that are registered periodically. For instance, these techniques can be applied to find a trend or a seasonality impact within monthly sales reported in tax returns. Popular techniques are ARMA, ARIMA, or Kalman filters.

(7) Anomaly detection. Anomaly detection aims to find unexpected observations or events that deviate significantly from normal patterns, see also chapters 4 and 6. In taxpayer supervision, these unusual patterns can lead to the detection of fraud, but anomaly detection can also be used to find unknown risks. Often anomaly detection proceeds by first modeling normal behavior (by applying predictive modeling techniques or cluster analysis) and subsequently defining a measure ('distance') of abnormality to identify anomalous observations. A classical technique in tax administrations and accounting is Benford's law.

(8) Recommendation systems. Recommendation systems recommend new products to customers based on the analysis of implicit or explicit preferences of these customers, reflected in their buying behavior or the ratings they give to products. This field has grown substantially with the rise of e-commerce. Novel techniques that can construct recommendation systems are collaborative filtering and matrix factorization [67, 63]. Another popular technique for constructing simple recommendation systems is the A-Priori algorithm [63]. Techniques from recommendation systems are not yet applied much in taxpayer supervision but could help improve taxpayer ser-

vices or gain insight in the combinations of risks.

(9) (social) Network analysis. Techniques from network analysis can be applied to extract information or risks from the (social) network of a taxpayer. In fraud detection these techniques are applied to the network of a fraudster, thus revealing new fraudsters. Network analysis is also applicable for analyzing social media or visualizing complicated legal structures [69, 63].

(10) Mining new data sources. Last decade, the machine learning community put considerable effort into extracting information from data sources that are coming from other sources then relation databases or surveys. Examples are collections of documents, images, webpages, twitter accounts, and recorded speech. Special techniques have been developed to tackle these new data sources [69, 63]. In taxpayer supervision, these techniques may be used for instance to find unregistered Internet companies.

Note that the (classes of) techniques above often require data pre-processing techniques, like data warehouse technology.

2.4.3 Matching supervision activities and analytical techniques

The classes of analytical techniques from section 2.4.2 can be mapped onto supervision activities of section 2.4.1, resulting in Table 2.4. The table is constructed by carefully questioning whether a class of analytical techniques can contribute to each supervision activity. This mapping is constructed based on practical experiences from the NTCA or known applications in related fields such as marketing or fraud detection.

Supervision activities and classes of analytical techniques	Descriptive statistics	Experimental design	Hypothesis testing	Predictive modeling	Cluster analysis	Time series analysis	Anomaly detection	Recommend. Systems	(social) Network Analysis	Mining new data sources
1. Risk Identification										
Horizon scans	X		X			X	X			X
Random audits	X	X	X							
Identify new risks from data	X					X	X			X
Segmentation of taxpayers	X			X	X					

Detecting fraud					X			X	X	
2. Risk Analysis										
Quantify risks with help of in-house or external data	X									
Hit rate scoring			X	X						
Random audits	X	X	X							
Tax gap estimations	X	X		X		X				
Trend analysis						X				
Root-cause analysis	X	X	X							
Estimating costs of treatment	X									
3. Prioritization										
Calculating human and other resources	X									
Optimizing Resource allocation	X			X						
4. Treatment										
Easy contacts	X			X	X			X		X
Desk audits	X			X			X			
Field Audits	X			X			X			X
Real-time checking of tax returns			X	X	X	X	X	X		
Pre-filled tax returns	X									
Administrating in the cloud				X			X			X
5. Evaluation										
Evaluation analysis			X	X						
Experimental design of evaluation		X								

Table 2.4: Mapping of (classes of) analytical techniques onto tasks of taxpayer supervision.

Descriptive Statistics is the most applicable class of techniques in taxpayer supervision, according to Table 2.4. The prominent role of descriptive statistics corresponds with practical experience where an initial, simple summary of raw data may already reveal significant insights. Predictive modeling ranks second. Predictive modeling techniques derive their strength in tax administrations from generalizing valuable information, available for a small group, to a much wider group. Think for instance about non-compliance information that is only known for a small number of audited taxpayers. A risk model may, based on this sample, predict the compliance of a much larger group of taxpayers.

2.4.4 A roadmap for analytics in compliance risk management

Davenport and Harris sketch five developmental stages of an analytical business: analytical impaired, localized analytics, analytical aspirations, analytical companies, and analytical competitors [32]. These stages are in no small degree recognizable for tax administrations, although tax administrations lack the competitive framework of businesses.

The first stage, 'analytically impaired', is characterized by businesses making decisions based on intuition only. Data is generally missing or of poor quality and not integrated. Analytical processes are lacking. Stage one is recognizable for some tax administrations where basic administrative processes of the government (company/citizen/property administration) are not in place yet, or data is not available in digital form. According to Davenport, Harris, and Morison [33] a business can overcome stage one by targeting 'low hanging fruit', i.e., identifying small-scale projects that show business potential. In taxpayer supervision, one may think about finding and testing basic audit selection rules for a risk for which data can be made available. Another possibility is acquiring and matching third-party data with data of the tax administration. At the taxpayer service side, one may start with registering and analyzing the type of questions that arise by taxpayers to get a better understanding of bottlenecks they experience.

The second stage, 'localized analytics', is characterized by autonomous analytical activity by individuals or disconnected teams within a business. Business-wide agreement on definitions is generally lacking, so 'multiple versions of the truth' may exist. In niches, however, isolated analysts might have achieved some excellent, tactical results. In tax administrations, that have many employees, many niches exist and setting up centralized policies takes time. This may be one of the reasons for the frequent occurrence of this stage among tax administrations nowadays. To overcome this stage, a strong effort from senior executives is needed to create a cohesive system of analytical activities [32].

The third stage, 'analytical aspirations', can be achieved by building business consensus around analytical targets, starting to build a business analytical infrastructure, create a business vision on analytics, target business processes that cross departments, and recruit analysts [33]. In the United Kingdom, The Netherlands, and some other countries, these transitional activities could be observed in 2016. The third stage is characterized by coordinated analytical objectives, separate analytical processes, analysts in multiple areas of business, early awareness, support of analytical possibilities among executives, and a proliferation of BI-tools. For taxpayer supervision, at least some activities mentioned in Table 2.4 have to be supported by analytics. By integrating external data, establishing business governance of technology and an analyt-

ical architecture, engaging senior leaders, working with main business processes, and developing relationships with universities and associations, the fourth stage can be reached.

The fourth stage is characterized by high-quality data, the presence of a Business Information plan, and the incorporation of analytical solutions in some business processes. Full executive support is in place, and change management is applied to build a fact-based culture. Most of the supervision activities of Table 2.4 are supported by analytics. Analytics brings insights to taxpayer supervision, and is structurally embedded in the compliance risk management strategy. In this stage, identification of compliance risks, analysis of trends, and root-cause analyses take place structurally, per segment of taxpayers, enabling a tax administration to match the results with the appropriate treatment.

The fifth stage is characterized by deep strategic insights, fully embedded analytical applications, highly professional analysts, a CEO with a passion for analytics, a broadly supported fact-based and learning culture, and a business-wide architecture. No tax administration has yet reached this stage, and it might not be the ambition of all tax administration to develop analytics to this extent.

The transition from the second stage to the third stage is relevant for most tax administrations. The Case Study below presents an initiative from 2014–2017, taken from the Netherlands Tax and Customs Administration.

2.5 Case Study: VAT refund risk model

The Netherlands Tax and Customs Administration (NTCA) receives numerous VAT refunds requests [14]. These requests, if approved, result in a payment of the NTCA to a taxpayer. All VAT refunds are automatically checked against risk rules to select risky VAT refund requests. If a VAT refund is risky, a manual inspection follows.

In 2014, the NTCA started a project aiming at replacing current risk rules — designed by domain experts — by a risk model, constructed by applying predictive modeling techniques. Both the old risk rules and the new risk model take advantage of domain knowledge and available data. The main difference is that with the old risk rules, hypotheses about risky features emerged in the minds of domain experts, and in the new risk model, a computer algorithm generates the hypotheses and subsequently tests them on historical data. The strength of the computer algorithm lies in its power to generate and check a vast amount of hypotheses on the historical data. Although many of the hypotheses generated by the computer algorithm may be of inferior quality compared to the hypotheses brought forward by domain experts, some hypotheses may outperform the old risk rules and only these are kept.

Before the project started, some significant developments had taken place at the NTCA. The government of the Netherlands approved a program to address structural issues in the operating model of the NTCA, called Investment Agenda (IA, [110]). The aim of the IA is providing the necessary response to changing taxpayer's expectations and significant technological developments. Within this context, a general trend towards centralization had started. Moreover, an awareness of the potential of analytics has spread among a small group of senior and middle management. The IA made it possible to invest in analytics in a time of budget cuts. Management created a small department ('Data & Analytics', now called 'Datafundamenten & Analytics') that started to realize 'data foundations' and initiated several projects, including the VAT refund project.

The VAT refund project consists of four stages; exploration phase, lab phase, pilot phase, and full implementation phase. A go/no go decision separates each phase. The project finished its full implementation phase successfully in 2017.

The 'exploration phase' aimed at estimating the financial benefits, the impact on processes, and the required changes in ICT. This phase was followed by the 'lab phase', that developed an operating risk model within three months. This first risk model showed promising results on historical data, but was not yet suitable to be applied in operations. In the 'exploration' and in the 'lab phase' approximately three analysts of the NTCA were involved.

After the 'lab phase', the 'pilot phase' followed, aimed at testing the risk model in practice. Two local tax offices were appointed as pilot-location. In the pilot phase, the development team was extended with VAT domain experts. Moreover, a small team of two professional programmers had been formed to streamline the initial code and to make it 'production-ready'. Finally, a pilot-support team of two employees was created to support the two pilot locations on the job floor.

It took three pilots, of three months each, to come to a final risk model delivering expected results. Each pilot refined the model further. For instance, after the first pilot it was noted that the selected tax returns had a high probability of containing an error, but on average a (too) low monetary value. At the end of the 'pilot phase', the regular ICT department became involved to develop a Workflow Management application, that is able to distribute the new risk signals efficiently to the desk auditors who handle the new signals.

As with most analytic projects, some unexpected side results were obtained. For instance, the riskiness of VAT refunds appeared to deviate substantially between the two pilot locations. This suggested shifting part of the workload from one location to the other.

2.6 Conclusions

Analytics seems to be a serious candidate for making taxpayer supervision more effective. Mapping the analytical techniques to the activities in the various stages of a Compliance Risk Management strategy shows the potential for taxpayer supervision in more detail. Nevertheless, analytics cannot support all activities, see Table 2.2. This observation leads to a hypothesis that supervision activities can be split into those that analytics can be improved with analytics and those that cannot. Our inventory Table 2.2 suggests that for taxpayer supervision, about half of the activities can be supported by analytics.

Although the OECD survey [40] gives practical examples of applying analytics for audit case selection, filing and payment compliance, taxpayer's services, debt management, and policy evaluation, the focus currently seems to lie on improving selection for tax auditing (higher hit rate, more revenue). The case study of Section 2.5 supports the idea that analytics can improve audit selection. However, there is a risk in paying too much attention to audit selection with analytics. Increasing attention for audit selection may implicitly shift the balance (between prevention and repression) needed in Compliance Risk Management from prevention to repression. This effect may in general occur since applications of analytics on the repressive side (e.g., audit selection) currently are more mature compared to applications on the preventive side (e.g., improving services). The effect may be canceled by putting more effort in developing preventive applications.

If we combine the five developmental stages of an analytical business: analytical impaired, localized analytics, analytical aspirations, analytical companies, and analytical competitors with the results of the OECD survey, it seems that most tax administrations are still in an early stage of development of applying advanced analytics for taxpayer supervision. The fact that some tax administrations state to apply analytics broadly, can probably be explained since the OECD survey did not make clear to what depth analytics are applied.

Analytics, in our opinion, does not achieve anything fundamentally new when it comes to the type of activities carried out by a tax administration. However, analytics could improve the foundation for a Compliance Risk Management strategy, leading to more rational decisions made by the management of a tax administration. Analytics, from that perspective, complements Compliance Risk Management. Especially when tax administrations succeed in using statistical techniques to draw predictions and make inferences about cause and effect, analytics will have an added value for Compliance Risk Management – influencing taxpayer behavior to comply with the rules. Before really confirming that analytics is more efficient and effective for taxpayer supervision, more proof is needed, and therefore, tax administrations are urged to

measure the impact of their (analytical) activities.

2.7 Discussion

A common misunderstanding is that analytical algorithms can solve business problems autonomously. According to Daniel Larose (2005: 4), this misunderstanding is partly caused by software vendors that, '... market their analytical software as being plug-and-play out-of-the-box applications that will provide solutions to otherwise intractable problems without the need for human supervision or interaction'. In reality, analytical experts are needed to guide the computer algorithms. Moreover, domain experts are crucial for drawing the right conclusions from the output of the techniques and to prevent automatic decision-making with far-reaching consequences for taxpayers. For instance, in risk identification, analytics does not come up with a fiscal risk directly. It mostly points to irregularities that *might* lead to a fiscal correction when studied by a domain expert. Therefore, it is essential to realize that analytics must support human experts and not vice versa.

An obvious limitation of analytical techniques is that one cannot get insights out of data that are not present in the data. For instance, insights cannot be extracted from data for new risks (e.g. risks related to new legislation). Moreover, available data may contain insufficient information to be 100% confident on a risk. More likely, the data contains clues, leading to an increased risk level, without providing certainty.

Privacy issues (as well as ethical issues) are of primal concern when applying analytics. At present, research is done to analyze data while preserving the privacy of individuals. This field is known as 'privacy preserving data mining'. Although some algorithms have been proved to preserve privacy, care should still be taken to manage the whole process adequately. More on privacy issues and analytics can be found in Haddadi et al [46].

3

Extending Logistic Regression Models with Factorization Machines

In this section we address the following research question:

> **Research Question** *Can we improve on current audit selection models by incorporating categorical variables with many values (like 'industry sector') that are valued by experts doing manual audit selection?*

Interest in including categorical variables with many levels was raised by a practical problem at the NTCA: a much used risk model did not contain these variables because they were rejected in a feature selection step. Nevertheless, experienced auditors used these variables successfully in the manual selection of audits. The idea originated that complementary techniques must be used to exploit the information hidden in these variables.

The research has value outside the realm of taxes as many data scientists encounter categorical features with many levels and are looking for the right way to incorporate them in risk models or other applications, as is evident from the questions on the subject on *www.stackoverflow.com*.

We focus on logistic regression models, a technique frequently used for risk modeling within tax administrations. Including categorical variables with many levels in a logistic regression model easily leads to a sparse design matrix. This can result in a big, ill-conditioned optimization problem causing overfitting, extreme coefficient values and long run times. Inspired by recent developments in matrix factorization, we propose four new strategies to overcome this problem. Each strategy uses a Factorization Machine that transforms the categorical variables with many levels into a few numeric variables that are subsequently used in the logistic regression model. The

application of Factorization Machines allows to include interactions between the categorical variables, often substantially increasing model accuracy. The four strategies have been tested on a data set of the NTCA and three public data sets, demonstrating superiority of the approach over other methods of handling categorical variables with many levels. The chapter is based on the following article:

- M. Pijnenburg and W. Kowalczyk. Extending logistic regression models with factorization machines. In *International Symposium on Methodologies for Intelligent Systems*, pages 323–332. Springer, 2017

3.1 Introduction

Logistic regression is a well-known classification algorithm that is frequently used by businesses and governments to model risks. However, logistic regression will run into problems when one tries to include categorical variables with many levels. The standard approach will transform each level of each categorical variable into a binary ('dummy') variable (see, e.g., [42]), resulting in a large, sparse design matrix. The sparsity usually leads to an ill-conditioned optimization problem, resulting in overfitting, extremely large values of model coefficients and long run times or even lack of convergence. The size and sparsity of the design matrix will increase even more if interactions are included between categorical variables with many levels or interactions between these categorical variables and some numeric variables. Finally, a large design matrix leads to a model with many coefficients, making it difficult to interpret.

Existing approaches for incorporating multi-level categorical variables into logistic regression reduce the problem of sparsity at the price of losing some information from data and consequently leading to models of inferior quality, see section 3.2.1 for an overview. This became clear to the authors when working on a risk model for selecting risky VAT tax returns for the Netherlands Tax and Customs Administration (NTCA), see also Section 2.5. Although the risk model performed pretty well, one experienced auditor was able to outperform the model by manually extracting information from categorical variables with many levels such as industry sector code and zip code. This information was clearly not picked up by the model, where standard approaches had been followed to include these categorical variables. The real-life example of the NTCA will be referred to in this chapter several times.

Logistic regression is traditionally used at the NTCA since it is a standard tool in the industry and the role of various features can be easily interpreted. Moreover, it performs well and its output can be interpreted as probabilities. This latter fact is important since it allows a decoupling of two key components of a tax return risk model: the probability of an erroneous tax return and the size of the financial loss connected to the error in the tax return, see [11].

In this chapter we propose four strategies for including categorical variables with many levels into a logistic regression model, by making use of Factorization Machines, [97]. Factorization Machines transform the categorical variables into vectors of numerical ones, taking interactions of the categorical variables into account. Factorization Machines can be viewed as an extension of matrix factorization methods, [58], that in turn have been developed in the context of recommendation systems, stimulated by the Netflix Challenge.

The chapter is organized as follows. Section 3.2 reviews the related literature. Section 3.3 introduces the four strategies of combining Factorization Machines and

Logistic Regression. Section 3.4 describes the experiments on four data sets. Section 3.5 contains conclusions and a discussion of the results.

3.2 Related Research

3.2.1 Existing Methods for Many Levels

A number of approaches have been suggested to deal with categorical variables with many levels. Many approaches focus on grouping levels of categorical variables into a smaller number of levels. This grouping can be done in a supervised manner (i.e. involving the target) or in an unsupervised way.

A well-known supervised way of grouping levels comes from the decision tree algorithm CART [21]. Here, levels are ordered by the percentage of cases in the target class. Subsequently, all levels with the percentage above a certain threshold value are grouped into one new level, and the remaining levels into another one. Breiman et al. [21] proved that this approach will find the optimal partitioning of a train set under the conditions that the target is binary, the new categorical variable has two levels, and a convex criterion (like Gini) is used to measure the quality of the partition. Several extensions of this approach have been developed, e.g. [24, 26], but they either lack the guarantee of finding the optimal partitioning, or have a substantially higher order of complexity.

Other, frequently employed, supervised ways of grouping levels include search methods, like forward or stepwise search [16]. Typically one starts with each level forming a group of its own. Then groups are merged one at a time based on various criteria, leading to fewer groups. This approach is used, for instance, in the CHAID algorithm [57].

A simple, but often effective, unsupervised way of grouping levels has been proposed by Hosmer and Lemeshow [53]. They suggest to group levels that occur infrequently. This approach gave the best results on our data sets from all supervised and unsupervised groupings tried. Another frequently used unsupervised approach is to let experts group the levels.

Other methods than grouping levels have been put forward. For example, in a data pre-processing step the categorical variable with many levels can be transformed into a numeric variable with help of an Empirical Bayes criterion, [77]. Another approach, see [10], is to find additional numeric predictors. For instance, if the categorical variable is “city”, the number of inhabitants of the city could be added to the data set as a predictor.

Although all these approaches for dealing with categorical variables with many levels have their merits, none of them was able to improve the current risk model at

the NTCA. So other approaches need to be considered.

3.2.2 Factorization Machines

Factorization Machines are introduced in a seminal paper of Rendle [97]. This class of models is often employed for recommendation systems, where categorical variables with many levels occur frequently. The model equation of a Factorization Machine is given by:

$$\hat{y} = w_0 + \sum_{j=1}^{s} w_j x_j + \sum_{j=1}^{s} \sum_{k=j+1}^{s} \langle \mathbf{v}_j, \mathbf{v}_k \rangle x_j x_k \tag{3.1}$$

where $\hat{y}$ is the predicted value for an observation with variables $x_1, \ldots, x_s$. The w's are numeric coefficients to be fitted and the $\mathbf{v}$'s are vectors to be fitted (one for each variable). All vectors $\mathbf{v}$ have the same (usually small) length r, which is an input parameter. These vectors can be interpreted as low dimensional numeric representations of levels.

The interesting part of equation (3.1) are the interaction terms. Instead of assigning a new coefficient w_{jk} to each interaction term, a Factorization Machine models the interaction coefficients as an inner product between the vectors $\mathbf{v}_j$ and $\mathbf{v}_k$. The introduction of such a vector for each variable reduces the number of interactions from $O(s^2)$ to $O(rs)$, so from quadratic to linear in the number of variables s. Typically, variables x_j in a Factorization Machine are binary variables resulting from transforming a categorical variable with many levels in dummy variables. In this case the number of coefficients is thus not quadratic in the number of levels, but linear. Note that when s is small (e.g., $s \leq 2r + 1$) there is no reduction in the number of coefficients.

The loss function that is used to find the optimal values of parameters in (3.1) usually involves a regularization term which controls the L^2 norm of model parameters. To find an optimum of the loss function several techniques can be used, among them Markov Chain Monte Carlo, Alternating Least Squares and Stochastic Gradient Descent [98].

3.3 Combining Logistic Regression with Factorization Machines

In this section we propose four new strategies for extending Logistic Regression with Factorization Machines. The key idea of these new approaches is the usage of Factorization Machines for squeezing relevant information from many-level categorical variables and their interactions into numeric variables and incorporating these latter in a logistic regression model. In this way, potential problems with large and sparse

design matrix are handled by Factorization Machines, while Logistic Regression takes care of combining "non-sparse" variables in a standard way.

Notation. We model a binary target y with help of p numeric variables $x_1, \ldots, x_p$, and q categorical variables $d_1, \ldots, d_q$ with $l_1, \ldots, l_q$ levels.

We will compare the performance of our four strategies with two benchmarks: (1) a logistic regression model without categorical variables, and (2) a logistic regression model where infrequent levels have been grouped as suggested by [53], see section 3.2.1. We start by introducing these latter methods in more detail.

3.3.1 Plain Logistic Regression (PLM)

The PLM model consists of a standard logistic regression model of the numeric variables $x_1, \ldots, x_p$. The model equation is:

$$\hat{y} = \frac{1}{1+e^{-z}}, \text{ where } z = \alpha_0 + \alpha_1 x_1 + \ldots + \alpha_p x_p. \tag{3.2}$$

The coefficients α_i are estimated by finding the unique maximum of the log-likelihood function over the train set.

3.3.2 Logistic Regression with Grouping (LRG)

This model groups infrequent levels of the categorical variables with many levels $d_1, \ldots, d_q$ into a default level for each d_j. The actual threshold for calling a level 'infrequent' depends on the size of the data set and can be found in Table 3.1. The grouping of infrequent levels leads to a new set of categorical variables $\tilde{d}_1, \ldots, \tilde{d}_q$ with less levels. Next, a standard approach is followed to replace categorical variables by dummy variables, i.e. each $\tilde{d}_j$ is transformed into $l_j - 1$ binary variables. Subsequently, PLM is applied on these binary variables and the numeric predictors.

In mathematical terms, the LRG model is given by:

$$\begin{aligned}\hat{y} = \frac{1}{1+e^{-z}}, \quad \text{where } z = {} & \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \\ & \alpha_{11} b_{11} + \ldots + \alpha_{1 l_1 - 1} b_{1 l_1 - 1} + \ldots + \alpha_{q l_q - 1} b_{q l_q - 1},\end{aligned} \tag{3.3}$$

where b's are binary variables. Note that in order to limit notation, we denote the coefficients of the logistic regression in all model equations ((3.2), (3.3), (3.4), (3.5), (3.6), and (3.7)) with $\alpha_0, \alpha_1, \ldots$, despite the fact that the values of these coefficients differ.

3.3.3 LRFM1

The model LRFM1 (Logistic Regression with Factorization Machines 1) is the first model showing our new approach. The categorical variables with many levels $d_1, \ldots, d_q$ are first put into a Factorization Machine f_0 whose coefficients are estimated from a train set with the target variable y. The output of f_0 — denoted by g_0 — is then added to the model equation of the logistic regression. Therefore,

$$\begin{aligned} g_0 = f_0(d_1, \ldots, d_q) \quad \text{and} \\ \hat{y} \quad = \tfrac{1}{1+e^{-z}}, \text{ where } z = \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \alpha_{g_0} g_0. \end{aligned} \tag{3.4}$$

3.3.4 LRFM2

Although LRFM1 is able to model interactions between categorical variables with many levels, it does not model interactions between the variables $d_1, \ldots, d_q$ and one or more numeric variables x_j. For this reason we allow the model equation (3.4) to be extended with additional variables $g_1, \ldots g_t$, where $t \leq p$. Each g_j is a prediction from a Factorization Machine f_j that takes as input the categorical variables $d_1, \ldots, d_q$ and a variable $\bar{x}_j$. The variable $\bar{x}_j$ is a discretized version of x_j, obtained by an equal frequency binning with 5 bins.

The coefficients of the Factorization Machine f_j are learned from a train set that contains the target y. Only variables g_j that significantly improve the results on the train set (compared to the model with only g_0, significance level $\alpha = 0.05$) will enter the model equation. Therefore, the model equation for LRFM2 is:

$$\begin{aligned} g_j = f_j(\bar{x}_j, d_1, \ldots, d_q) \text{ and } \hat{y} = \frac{1}{1+e^{-z}}, \text{ where} \\ z = \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \alpha_{g_0} g_0 + \alpha_{g_1} g_1 + \ldots + \alpha_{g_t} g_t. \end{aligned} \tag{3.5}$$

3.3.5 LRFM3

Instead of learning the coefficients of a Factorization Machine f on a train set with known binary target y, we can do an intermediate step. We first fit a logistic regression model with the numeric variables $x_1, \ldots, x_p$ on the train set (so without $d_1, \ldots, d_q$), and then compute the deviance residuals r_i (see [53]):

$$r_i = \pm \sqrt{2 \left[y_i \log \frac{y_i}{\hat{y}_i} + (1 - y_i) \log \frac{1 - y_i}{1 - \hat{y}_i} \right]},$$

where $\hat{y}_i$ denotes the predicted probability that $y_i = 1$ and the sign is $+$ iff $y_i = 1$. The residual vector $\mathbf{r}$ can then be used to train the coefficients of the Factorization Machine instead of the original target $\mathbf{y}$. This will give a Factorization Machine $\tilde{f}$.

Note that the Factorization Machine is now performing a regression task, instead of classification. LRFM3 is described by the equations:

$$\begin{aligned} h_0 &= \tilde{f}(d_1, \ldots, d_q) \text{ and} \\ \hat{y} &= \tfrac{1}{1+e^{-z}}, \text{ where } z = \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \alpha_{h_0} h_0. \end{aligned} \tag{3.6}$$

3.3.6 LRFM4

Similarly as LRFM1 was extended to LRFM2, we can extend LRFM3 to LRFM4. More specifically, we form additional variables h_j by including a discretized numeric variable $\bar{x}_j$ in the Factorization Machine that is trained on residuals. Only variables h_j that significantly ($\alpha = 0.05$) improve the result on the train set will enter the model equation. This provides our last strategy:

$$\begin{aligned} h_j &= \tilde{f}(\bar{x}_j, d_1, \ldots, d_q) \text{ and } \hat{y} = \tfrac{1}{1+e^{-z}}, \text{ where} \\ z &= \alpha_0 + \alpha_1 x_1 + \ldots \alpha_p x_p + \alpha_{h_0} h_0 + \alpha_{h_1} h_1 + \ldots + \alpha_{h_r} h_r. \end{aligned} \tag{3.7}$$

3.4 Experiments

Interest in including categorical variables with many levels was raised by a practical problem at the NTCA. Most research has been focused on this data set. However, in order for the research to be reproducible, we also applied our approach to three public data sets in the UCI Repository [65] that contain categorical variables with many values. See Table 3.1 for some key characteristics of the four data sets. We considered a variable to have 'many levels' if the number of levels exceeded 30. This number corresponds roughly with the situation where the design matrix becomes sparse in our four data sets. Below we will describe the data sets, the exact parameters of the experiments, and the results.

3.4.1 Data Sets

3.4.1.1 Tax Administration

This data set consists of approximately 80,000 audited VAT tax returns. A small part of these tax returns (17.5%) were found to contain one or more erroneous statements when audited. The data set has 33 numeric variables that are the result of a stepwise feature selection process that started with over 500 variables.

	Data Set			
	tax	kdd98	retail	census
# observations	86,235	95,412	532,621	199,523
# numeric vars	33	18	3	23
% target = 1	17.5%	5.0%	29.1%	6.2%
% target = 0	82.5%	95.0%	70.9%	93.8%
threshold infreq. level (LRG meth.)	100	100	1000	100
cat. features (# levels)	zipcode (1,027) industry sector (3,747)	DMA (207) RFA 11 (101) RFA 14 (95) RFA 23 (87) OSOURCE (869) ZIP (19,938)	InvoiceNo (25,900) Description (4148) CustomerID (4,373) Cntr (38)	ind. code (52) occ. code (47) prev. state (51) hh. stat (38) cntr fthr (43) cntr mthr (43) cntr birth (43)

Table 3.1: Summary of data sets used in the logistic regression / Factorization Machine experiments

3.4.1.2 KDD 98 cup

This data set comes with a binary target and a numeric target. We only use the former as we focus on classification. Additionally, we only used the 'learning' data set and not the 'validation' data set. The original data set contains 480 variables. We selected 22 variables to get a data set similar to the tax data. The variable selection has been done by keeping the variables reported in [44] (page 147, Table 6) and adding the two variables with many levels: OSOURCE and ZIP. Missing values have been replaced by 0, except for WEALTH where the median has been inserted.

3.4.1.3 Online Retail

The following data processing steps have been performed: (1) canceled transactions have been removed, (2) InvoiceDate has been split in a date part and a time part, (3) StockCode has been removed since its values can be mapped almost one-to-one to the values of Description. The data set does not contain a target. We created a binary target by defining the target to be 1 if the variable Quantity is larger or equal to 10, and 0 otherwise. After this, Quantity has been removed from the data set.

3.4.1.4 Census

The following data processing steps have been taken: (1) Weight has been discarded as is advised in the data description, (2) variables that are aggregates of other variables have been removed. For instance, Major Industry Code aggregates the levels of Industry Code. For this reason the variables: Major Industry Code, Major Occupation Code, Previous Region, Household Summary, Migration Code Reg, Migration Sunbelt, and Live1year House have been removed. Similarly, (3) Veteran Questionnaire is removed since most relevant information is in Veteran Benefits. Finally, (4) Migration Code MSA has been removed since it is highly collinear with Previous State.

3.4.2 Model Quality Measures

Various performance measures can be used to assess the results of a classification algorithm (e.g. accuracy, precision, area under the curve, recall). At the NTCA audit capacity is limited and rather fixed. For this reason, one is more interested in using the available capacity to the best (so avoiding false positives), than trying to find all taxpayers that make an error (avoiding false negatives). *Precision* (i.e. the true positives divided by the sum of true positives and false positives) is therefore a natural measure, and more useful than accuracy (the sum of true positives and true negatives divided by all cases). We applied precision at a '10% cut-off level', measured on a test set (i.e., we select the 10% highest scoring observations of a test set and then compute the precision), in similar fashion as is done at the NTCA.

For reference, we have also provided the frequently used Area Under the Curve (AUC) characteristic. For confidentiality reasons the precision and AUC could not be reported for the tax data set (although tested in practise). Instead we have provided the *increase in precision*, measured using the Plain Logistic Regression as a baseline. For instance, if the precision of Plain Logistic Regression is 60%, and the precision with the new technique is 66%, then the *increase in precision* is $100\% \cdot (66 - 60)/60 = 10\%$.

In our experiments we used 5-fold cross validation to get reliable estimates of all quality measures listed above.

3.4.3 Settings Factorization Machines

In our experiments Factorization Machines were constructed with libFM software [98] with the following settings: number of iterations: 25, lengths of the parameter vectors $\mathbf{v}$: 16 (see equation (3.1)), and the optimization technique is set to 'MCMC'. The standard deviation of the normal distribution that is used for initializing the parameter vectors $\mathbf{v}$ in MCMC is set to 0.1. When building the Factorization Machines in

	Precision (%)			Precision (% increase w.r.t. PLR)				AUC		
	kdd98	retail	cens.	tax	kdd98	retail	cens.	kdd98	retail	cens.
PLR	6.05	69.3	42.7	0.0%	0.0%	0.0%	0.0%	.5348	.7845	.9358
LRG	7.88	80.7	44.7	2.7%	30.3%	16.4%	4.6%	.5748	.8442	.9439
LRFM1	7.92	99.6	44.5	3.9%	31.0%	43.7%	4.2%	.5775	.9634	.9426
LRFM2	7.92	99.8	44.5	5.1%	31.0%	44.1%	4.2%	.5775	.9689	.9426
LRFM3	6.93	99.5	44.3	6.1%	14.6%	43.6%	3.7%	.5502	.9710	.9410
LRFM4	6.79	99.5	44.3	9.6%	12.3%	43.6%	3.7%	.5553	.9713	.9410

Table 3.2: Performance (measured in precision, increase of precision with relation to Plain Logistic Regression, and Area Under Curve) for each strategy on all data sets using five-fold cross-validation. Absolute values for the tax data set have been deliberately omitted for confidentiality reasons.

approaches LRFM1 and LRFM2 we set the 'task' to 'classification', and in the remaining two cases to 'regression'.

3.4.4 Results

The results of applying two benchmark methods Plain Logistic Regression (PLR) and Logistic Regression with Grouping (LRG), as well as our four strategies are summarized in Table 3.2. The columns *precision* and *AUC* are not filled for the tax administration data set, because of confidentiality reasons.

3.5 Conclusion and Discussion

Looking at Table 3.2, some conclusions can be drawn. First, our proposed strategies give better results than plain logistic regression for all data sets. Second, when comparing with LRG (the strategy that gave the best results from the methods of section 3.2.1), we see a subtler picture. Our methods outperform LRG clearly on the tax data and the retail data. For the tax data we see that taking interactions of the categorical variables with numeric variables into account, while training on residuals (i.e. LRFM4), can substantially improve the result. When looking at the data set kdd98, the approaches that train directly on the target y (LRFM1 and LRFM2) are able to give slightly better results compared to LRG. However, LRG gives a slightly better result for the census data set. The latter might be caused by the relatively small number of levels of the categorical variables (maximum 52). Finally, our four strategies LRFM1, LRFM2, LRFM3, LRFM4 lead to different results on different data sets, without one

strategy being the best for all data sets. The only exception is that in some cases the result of LRFM2 equals LRFM1 or the result of LRFM4 equals LRFM3. This is the case when no interaction term exceeded the significance level for entering the modeling equation. Therefore, we suggest to explore all four strategies in a practical problem setting.

The results of this chapter show that Factorization Machines can be successfully combined with Logistic Regression to overcome problems with categorical variables with many levels. This will lead to better performing (risk) models. We think that our methods can be adjusted without much effort to allow inclusion of categorical variables with many levels in other classification algorithms that suffer from a sparse, ill-conditioned model matrix, like other Generalized Linear Models or Support Vector Machines. Also a generalization to a multinomial logistic regression is straightforward. Note that some well-known classification algorithms have problems with categorical variables with many levels. For example, the standard implementation in R of RandomForest [64] accepts only categorical variables with at most 53 levels.

Further research can address the issue of explainability of a Factorization Machine. Although our strategies lead to relatively simple logistic regression models, the introduction of the Factorization Machines worsens the explainability for that part of the model. We think that this problem can be solved by applying various dimensionality reduction and visualization techniques to the matrix $\mathbf{V}$ that consists of vectors $\mathbf{v}$ that represent levels of categorical variables, [107]. One could experiment as well with using an L^1 norm as a regularization term in the Factorization Machine.

Finally, we mention that some categorical variables with many values, like zip codes, may have a relation with ethnicity. For example people with the same ethnic background might be clustered in certain zip codes. Since the NTCA wants to avoid ethnic profiling, it wants to investigate this issue prior to including zip codes in a risk model.

4

Singular Outliers: Finding Common Observations with an Uncommon Feature

In this chapter we focus on the research question:

> **Research Question** *Can we develop a new anomaly detection algorithm tailored to find the outliers that are of interest to a tax administration?*

Firstly, we looked carefully at the types of outliers that are of interest for tax administrations. This leads us to define the concept of *singular outliers*. Singular outliers are multivariate outliers that differ from conventional outliers by the fact that the anomalous values occur for only one feature (or a relatively small number of features). Subsequently, we developed a new algorithm (Singular Outlier Detection Algorithm – SODA) for detecting these outliers. The SODA algorithm is applied successfully to tax data. In order to obtain reproducible findings, the algorithm is applied as well to five public, real-world data sets and the outliers found by it are qualitatively and quantitatively compared to outliers found by three conventional outlier detection algorithms, showing the different nature of singular outliers. The chapter is based on the following article:

- M. Pijnenburg and W. Kowalczyk. Singular outliers: Finding common observations with an uncommon feature. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 492–503. Springer, 2018

4.1 Singular Outliers

Currently, many outlier detection algorithms exist [25]. However, when trying to find tax fraud, we found that hardly any of these is able to detect a particular type of outliers, which we will call *singular outliers*. In this chapter we will describe these singular outliers and present a newly developed algorithm (Singular Outlier Detection Algorithm – SODA) that can find these anomalies.

Roughly speaking, singular outliers are observations that show an anomalous value for *one* feature (or a relatively small set of features), while displaying common behavior on all other features. This feature with the anomalous value will be called the *discriminating feature*. The discriminating feature may be different for each outlier and finding the discriminating feature is part of the learning process. Singular outliers are typically overlooked by current outlier detection algorithms that prefer observations with anomalous values on as many features as possible.

To explain the concept of singular outliers more clearly, suppose we have a set of points $\mathbf{X}$ in a 5-dimensional vector space (with the standard metric) and a particular point $\mathbf{x} \in \mathbf{X}$, surrounded by its 20 nearest neighbors. Denote the mean vector of the 20 neighbors with $\mathbf{m}$, i.e., $\mathbf{m} = 1/20 \sum_1^{20} \mathbf{n}_i$. This allows calculating the absolute distances between $\mathbf{x}$ and $\mathbf{m}$ for each dimension. When visualized in a needle plot, the result may look as one of subplots of Figure 4.1. The subplot in the left shows large deviation from $\mathbf{m}$ in all dimensions. This observation can be labeled a conventional outlier. However, the right subplot shows an observation with small deviations to $\mathbf{m}$ on all but one dimension (dimension 4). This is what we will call a singular outlier.

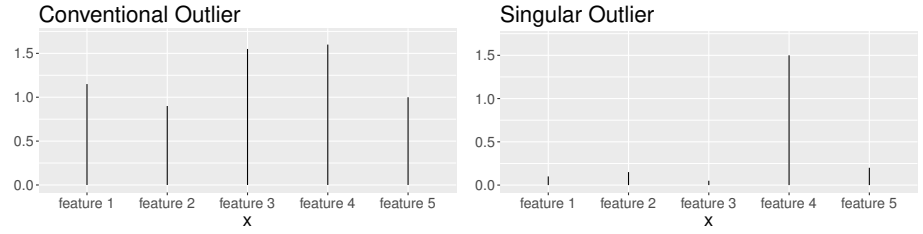


Figure 4.1: Needle plots of the absolute differences $|x_i - m_i|$ ($i \in 1, \ldots, 5$) of a point $\mathbf{x}$ and the center $\mathbf{m}$ of its 20 nearest neighbors.

Singular outliers can provide interesting insights in the field of fraud detection or data quality. We will illustrate this by our experiences at the Netherlands Tax and Customs Authority (NTCA), where we were asked to contribute to the selection of erroneous VAT tax returns for field audits. It turned out that tax returns containing many unusual values, are often less risky than tax returns containing only one (or two) anomalous fields. This apparent contradiction can be explained partly by efforts

of taxpayers to conceal tax evasion. But also by the existence of unconventional businesses with non-standard business models, that produce uncommon values on many fields of a tax declaration, without any increased risk of tax evasion. Moreover, it was noted at the NTCA that taxpayers sometimes unintentionally change two adjacent fields, leading to an example of singular outliers in the field of data quality. The examples in section 4.4 point to other application areas.

Singular outliers usually differ from those found by conventional outlier detection algorithms. Let us consider a real life example that will be more elaborate treated in section 4.4. In the example two algorithms, the newly developed SODA-algorithm and the popular LOF algorithm, [22] are used to find outliers in a data set containing the marks on 5 courses of 88 students[74]. Figure 4.2 shows two identical parallel coordinates plots (for more information on parallel coordinates plots see e.g., [23]) of this data set, but each plot highlights different outliers. The left plot highlights three singular outliers. We see that the singular outliers are students that have common marks on at least three subjects, but one or two exceptional marks. In contrast, the right subplot shows the same parallel coordinates plot but with three *conventional* outliers highlighted. These students have exceptional marks on all subjects (in this case exceptionally high). The singular outliers are interesting, since valuable insight might be gained in exploring the cause of the one (or two) exceptional mark(s) of the students.

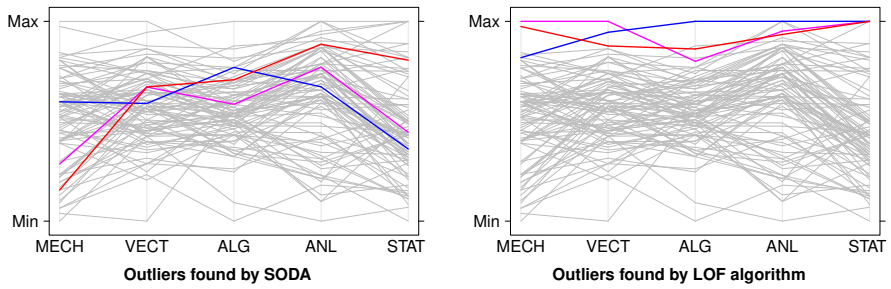


Figure 4.2: Two identical parallel coordinates plot of the Marks data set. The left plot highlights three observations that are singular outliers (detected by the SODA algorithm). The right plot highlights three conventional outliers (detected by the LOF algorithm). We see that the singular outliers have one or two features with exceptional values, whereas the conventional outliers show exceptional values on all features.

We will now define formally a singular outlier.

Definition 1 *An observation is called a* singular outlier *if there exits one feature (or a relatively small number of features), called the* discriminating feature(s) *such that the*

observation is an outlier when the discriminating feature is taken into account, but no outlier when the features are restricted to the non-discriminating features.

The purpose of this chapter is to point to the class of singular outliers and present an algorithm for finding them. The chapter is organized as follows. Section 4.2 gives a short overview of classes of outlier detection algorithms. Subsequently Section 4.3 describes the SODA algorithm for finding singular outliers and explains the experimental setup to test the algorithm on five public data sets. Section 4.4 contains the results of the experiments. Finally, Section 4.5 contains the conclusions of the chapter and a discussion of the results.

4.2 Related Work

Chandola, Banerjee, and Kumar [25] present an overview of commonly used outlier detection techniques. They distinguish five classes of unsupervised outlier detection algorithms: nearest neighbor-based algorithms (including density based approaches), clustering-based algorithms, statistical algorithms, information theoretic algorithms, and spectral algorithms. Goldstein and Uchida [43] present a similar categorization.

Nearest neighbor and clustering-based outlier detection are the most used categories in practice, according to Goldstein and Uchida. Of these two categories, nearest-neighbor based algorithms perform better in most cases [43]. Moreover it is useful to split the class of nearest neighbor algorithms in two subclasses: *local* algorithms and *global* algorithms. In the remaining categories outlier detection algorithms, the statistical algorithm HBOS (Histogram-based Outlier Score) performs remarkably well in experiments [43].

We will briefly describe three algorithms that we will compare with SODA: Local Outlier Factor, k^{th}-Nearest Neighbor and HBOS.

The Local Outlier Factor (LOF) is introduced by Breunig et al. [22]. The basic idea is to compare the density of an observation $\mathbf{x}$ with the density of its k closest neighbors $N^k(\mathbf{x})$. If we ignore a small subtlety that may arise if the k^{th}-neighbor of a point is not unique, the LOF score can be calculated simply by,

$$LOF_k(\mathbf{x}) = \frac{1}{k} \sum_{o \in N^k(\mathbf{x})} \frac{d_{Eucl}(\mathbf{x}, \mathbf{x}_{(k)})}{d_{Eucl}(\mathbf{o}, \mathbf{o}_{(k)})}, \tag{4.1}$$

where $d_{Eucl}(\mathbf{x}, \mathbf{y}) = \|\mathbf{x} - \mathbf{y}\|$ is the Euclidean distance.

The k^{th} nearest neighbors outlier detection algorithm is straightforward: the distance to the k^{th} neighbor is used as an anomaly score [96]. By taking the distance to the k^{th} neighbor instead of the average distance to the k^{th} nearest neighbors, the algorithm is better able to detect a small cluster of outliers.

The Histogram Based Outlier Score [43] starts with making a histogram for each feature in the data set. Then, for each observation, the height of the bins it resides are multiplied. This will result in a positive number. Subsequently the negative of this number is taken as an outlier score. In the experiments, the frequently used Sturges' formula is applied to compute the number of bins.

4.3 Singular Outlier Detection Algorithm

4.3.1 The Algorithm

We propose an algorithm called SODA (Singular Outlier Detection Algorithm) to find singular outliers. The algorithm involves several steps which are specified in Algorithm 1. The input of the algorithm is a data set with n observations and p numeric features as well as a parameter k that specifies the number of nearest neighbors. The output of the algorithm is an outlier score for each observation; large scores represent outliers. Additionally, the discriminating feature for each observation is given as output. The latter is a convenient starting point when manual inspection of outliers is required.

In the first step, for each observation $\mathbf{x_i}$, $i = 1, \ldots, n$, its k neighbors N_i^k are found using the Manhattan metric (also called the 'distance based on the L_1 norm'). Subsequently, the center of these neighbors, $\mathbf{m_i}$, is determined as the trimmed mean, i.e., the mean calculated after removing the largest value and the smallest value along each dimension. These extreme values are ignored to limit their impact on $\mathbf{m_i}$.

As a next step, the Euclidean distance $d_{Eucl}(\mathbf{x_i}, \mathbf{m_i})$ as well as the Manhattan distance $d_{Manh}(\mathbf{x_i}, \mathbf{m_i})$ are calculated. We will call the ratio of these two distances LEMR (*Local Euclidean Manhattan Ratio*), i.e.:

$$\text{LEMR}(\mathbf{x_i}) = \frac{d_{Eucl}(\mathbf{x_i}, \mathbf{m_i})}{d_{Manh}(\mathbf{x_i}, \mathbf{m_i})}. \tag{4.2}$$

This ratio is the 'singular outlier score' for observation $\mathbf{x_i}$. It is an indicator of the 'spread' of the values of the vector $\mathbf{v_i} = \|\mathbf{x_i} - \mathbf{m_i}\|$. The left subplot of Figure 4.3 displays a 2-dimensional example with the Euclidean unit sphere (i.e., the circle) as well as the Manhattan unit sphere (the square: all points with a Manhattan distance 1 to the origin). It is clear from the figure that points on the coordinate axes (i.e., singular outliers) are in both spheres, so the ratio of the Euclidean to the Manhattan distance is 1. On the other hand, points that are on the diagonals (the opposite of singular outliers) have the largest difference in Euclidean and Manhattan distance to the origin and consequently have a low LEMR value. The right subplot of Figure 4.3 shows the LEMR value in this 2-dimensional example.

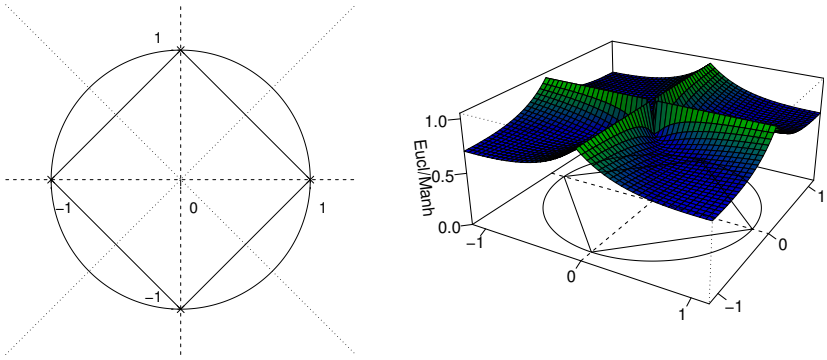


Figure 4.3: Left: the Euclidean unit sphere and the Manhattan unit sphere in 2 dimensions. Points on coordinate axes (singular outliers) have a maximal value of the Local Euclidean Manhattan Ratio (LEMR = 1). Points on the diagonals (i.e., conventional outliers) have a minimal LEMR value (LEMR $=1/\sqrt{2}$), see subplot right. In more dimensions the difference between the Euclidean and the Manhattan distance is more pronounced.

The value of LEMR cannot exceed 1 since the Euclidean length of a vector $\|\mathbf{v}\|$ is always smaller or equal to the L_1-norm $\|\mathbf{v}\|_1 = \sum |v_j|$. The LERM value of 1 occurs when all but one components of $\mathbf{v}$ equal 0. The value of LEMR is a minimum in case the elements of $\mathbf{v}$ are all of equal length (and unequal 0). The minimal value depends on the number of features p and is given by $1/\sqrt{p}$.

The calculation of the nearest neighbors in the first step of the algorithm is performed using the Manhattan distance in contrast to the more frequently used Euclidean distance. The reason is that the discriminating feature of a singular outlier ideally deviates strongly from other observations. This deviation may however be so large that it can have too big influence on the determination of the nearest neighbors. In the worst case, this would lead to neighbors that only share a similar value for the discriminating feature. To diminish the effect, we prefer the Manhattan distance that is less influenced by extreme values in one feature.

The algorithm has one parameter k. A (too) small value of k will lead to observations $\mathbf{x}$ with very few neighbors. Consequently, it might result in outliers whose non-discriminating features have a risk of being not so common. A (too) large value of k usually has less severe consequences. However, in theory it can lead to bad results if the data points are gathered in many small clusters and k exceeds the cluster size. For instance this may happen if the data contains binary features. In our experiments we used $k = n/5$.

The time complexity of the singular outlier detection algorithm is $\mathcal{O}(n^2pk)$.

Algorithm 1: singular Outlier Detection Algorithm (SODA)

Input : A data set $\mathbf{X} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots\}$ with p (numeric) features,
k the number of nearest neighbors

Output: 1) outlier score. The observations with the largest scores represent the singular outliers,
2) discriminating feature for each $\mathbf{x}_i$. This indicates the feature that contains the anomalous value.

1 Find the k nearest neighbors N_i^k for each observation $\mathbf{x_i}$ in $\mathbf{X}$ based on the Manhattan distance d_{Manh}.
2 **for** *each observation (row)* $\mathbf{x}_i$ *in* $\mathbf{X}$ **do**
3 Compute the vector of trimmed means $\mathbf{m}_i$ of the neighbors N_i^k:
$\mathbf{m_i} = \underset{\mathbf{x}_j \in N_i^k}{\text{mean}_{trimmed}}\{\mathbf{x}_j\}$
4 Compute
5 $d_{Eucl}(\mathbf{x}_i, \mathbf{m}_i) = \sqrt{\sum_{s=1}^{p}(x^s - m^s)^2}$
6 $d_{Manh}(\mathbf{x}_i, \mathbf{m}_i) = \sum_{s=1}^{p}|x^s - m^s|$
7 outlier score$(\mathbf{x}_i) = LEMR(\mathbf{x}_i) = \frac{d_{Eucl}(\mathbf{x}_i, \mathbf{m}_i)}{d_{Manh}(\mathbf{x}_i, \mathbf{m}_i)}$
8 discriminating feature$(\mathbf{x}_i) = \underset{s \in \{1,\ldots,p\}}{\text{argmax}}\,|x^s - m^s|$
9 **end**

4.3.2 Measurement of Characteristics of Singular Outliers

To determine whether an outlier can be called *singular*, two characteristics must hold. These two characteristics follow from definition 1 and are listed below.

No outlier with respect to non-discriminating features When the discriminating feature is removed, the observation stops to be an outlier.

Outlier with the discriminating feature The observation must become an outlier when the discriminating feature(s) is taken into account.

To quantify these two qualitative characteristics, we will formulate two measures. These measures will be applied in section 4.4, Table 4.2 to see whether the outliers found by SODA can be called singular outliers.

The first characteristic is measured for an outlier by removing the discriminating feature from the data set and then calculate an outlier score. This outlier score must be low for the characteristic to be valid. In principle any outlier score can be used. In this chapter the LOF score is chosen, since it is a well established outlier score that performs well on a broad variety of data sets [43]. A LOF score around 1 will be accepted as a sign that the first characteristic is applicable.

The second characteristic is measured for an outlier by computing an outlier score with and without the discriminating feature. Subsequently the ratio of the outlier scores is taken as a measure. Again, in principle any outlier score can be taken, but in this chapter we choose the LOF score. Hence, the measure of the second characteristic becomes $\frac{\text{LOF}_{all}(\mathbf{x})}{\text{LOF}_{w.o.discr}(\mathbf{x})}$. A large ratio signifies that the addition of the discriminating feature has substantially increased the outlier score (LOF) and is taken as a sign that the second characteristic is applicable.

In section 4.4 the outliers found by SODA will be compared with outliers of the three conventional outlier algorithms mentioned in section 4.2. In order to apply the two measurements above, we have to determine the discriminating feature for outliers found by these algorithms as well. For the algorithms based on nearest neighbors (LOF and k^{th}NN) a natural choice is to take:

$$\text{discriminating feature}(\mathbf{x}) = \underset{s\in\{1,\ldots,p\}}{\text{argmax}} |x^s - x^s_{(k)}|, \tag{4.3}$$

where $x^s_{(k)}$ is the s^{th} feature of the k-nearest neighbor of $\mathbf{x}$. For the HBOS algorithm we take the feature with the lowest bin height as the most discriminating feature.

4.4 Comparison

This section compares the outliers found by SODA with outliers detected by three conventional outlier detection algorithms (mentioned in section 4.2) to five real world data sets, that will be described shortly. The comparison consists of parallel coordinates plots and the two measurements described in section 4.3.2.

The selected data sets are listed in Table 4.1, along with some key properties. One of these properties is the value k that is used in the SODA algorithm and the two conventional outlier detection algorithms based on nearest neighbors (LOF and k^{th}NN). In the experiments, k is set to (approximately) one-fifth of the number of observations, see section 4.3.1.

data set name	# rows	# columns	k
Marks	88	5	20
Istanbul Stock Exchange	536	9	100
Wholesale Customers	440	6	80
Polish Bankruptcy	5,907	5	1,000
Algae	306	8	60

Table 4.1: Data sets used in the singular outlier experiments with key indicators.

Measurement 1 - no outlier without discriminating feature: LOF_{-d}					
	data sets				
	Marks	Istanbul	Wholesale	Polish	Algae
algorithm					
SODA	1.02	0.99	1.16	1.01	1.01
LOF	1.45	2.57	2.98	147.7	3.84
k^{th}NN	1.42	2.56	3.69	147.7	4.20
HBOS	1.39	2.57	4.02	122.0	3.28
Measurement 2 - outlier with discriminating feature: $\frac{\text{LOF}_{all}}{\text{LOF}_{-d}}$					
	data sets				
	Marks	Istanbul	Wholesale	Polish	Algae
algorithm					
SODA	1.09	1.28	2.31	5.75	1.99
LOF	1.04	1.08	2.11	1.46	1.73
k^{th}NN	1.02	1.04	1.60	1.46	1.27
HBOS	1.06	1.06	1.06	1.13	1.08

Table 4.2: Measurements of the two characteristics of singular outliers (see section 4.3.2) for the outliers selected by SODA, LOF, k^{th}NN and HBOS on five data sets. Each algorithm is allowed to pick 3 outliers and the values reported are averages over these three outliers. The first measurement computes an outlier score (LOF) without the discriminating feature. The second measurement computes the ratio of the LOF score with all features and the LOF score without the discriminating feature. Singular outliers are characterized by a value close to 1 for measurement 1 and a large value for measurement 2. We see that the outliers selected by SODA can be termed 'singular outliers' and differ on these characteristics from the outliers found by conventional outlier detection algorithms.

4.4.1 Marks Data

The first data set, *Marks*, comes from Mardia, Kent and Bibby [74] and consists of the examination marks of 88 students in the five subjects mechanics, vectors, algebra, analysis and statistics. We used these unstandardized marks.

The parallel coordinates plot of this data set is displayed in the introduction, see Figure 4.2. Clearly, the SODA algorithm picks up students that perform around average for most subjects, except for one subject. In contrast, the LOF algorithm selects students that perform exceptionally well on all subjects. The measurements of Table 4.2 confirm that the SODA outliers can be called singular outliers: the average value for the SODA outliers on measurement 1 equals 1.02, which is close to 1 and much

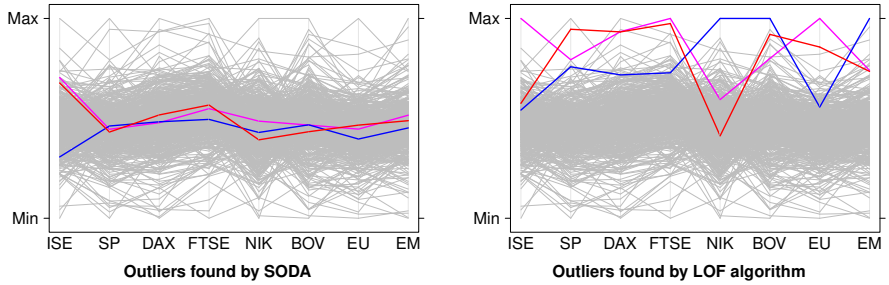


Figure 4.4: Parallel coordinates plot for the Istanbul Stock Exchange data set with the three largest outliers highlighted.

lower compared to the values of other algorithms. The average value on the second measurement (1.09) is not very large, but at least larger than the values for the outliers found by the other algorithms.

4.4.2 Istanbul Stock Exchange Data

The Istanbul Stock Exchange data set [3] displays the daily increase in eight major stock exchange indices for the period January 5 2009 to February 22, 2011. The data are already standardized.

The parallel coordinates plots of Figure 4.4 show that the SODA algorithm has picked three days where the average stock return is common (0.3 %), but the Istanbul stock exchange performed deviantly. In contrast, the LOF algorithm has selected three days on which most stock indices got exceptional good returns (average return of these days was 4.1 %, average of all days is less than 0.1 %). Table 4.2 confirms the differences of the outliers found by SODA and the three conventional algorithms.

4.4.3 Wholesale Customers Data

The Wholesale Customers data set [1] contains the annual spending on six product categories for 440 customers of a wholesaler in Portugal. These not standardized spending amounts were input for the algorithms. It is clear from the parallel coordinates plots in Figure 4.5 that the SODA algorithm selects customers with average sales on all product categories, but one. The LOF algorithm selects customers with exceptional high purchases on at least three product categories.

The statement that SODA picks out singular outliers is confirmed by looking at the measurements of Table 4.2. However, some experts might be inclined to call the red and / or pink line of the LOF algorithm singular outliers as well. See Section 4.5 for

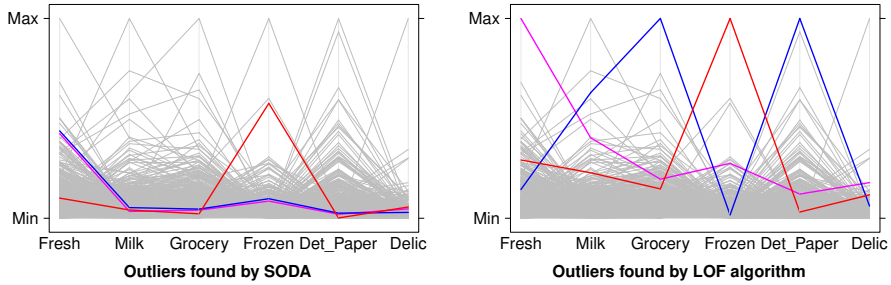


Figure 4.5: Parallel coordinates plot for the Wholesale Customers data set with the three largest outliers highlighted.

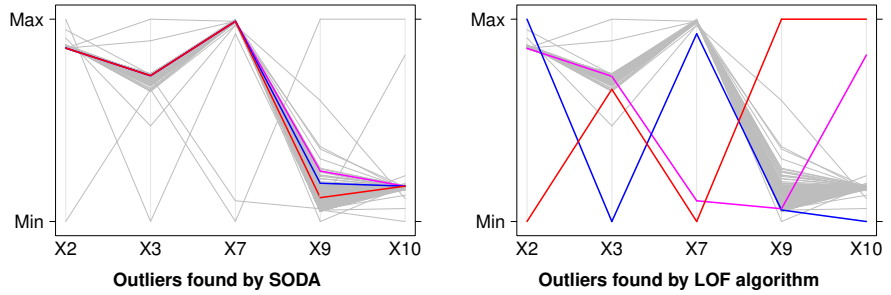


Figure 4.6: Parallel coordinates plot for the Polish Bankruptcy data set with the three largest outliers highlighted.

a discussion of this aspect.

4.4.4 Polish Bankruptcy Data

The Polish Bankruptcy data contains financial information of Polish companies and was originally used to model the probability of a bankruptcy [117]. We took the 'five year set' and standardized the ratios. We selected five financial ratios that correspond to major economic indicators of the company: X2 (total liabilities / total assets), X3 (working capital / total assets), X7 (Earnings Before Interest and Taxes / total assets), X9 (sales / total assets), and X10 (equity / total assets). After removing observations with missing values, 5,907 observations remain.

Comparing the outliers found by SODA and LOF in the parallel coordinates plots of Figure 4.6, we see that the SODA outliers show more common behavior than the LOF outliers. It is not clear from the figure whether the SODA outliers have at least one anomalous value. The latter is confirmed by Table 4.2. The first measurement shows

an average LOF value of 1.01 for the SODA outliers, indicating that the observations are not considered outliers without the discriminating feature. Simultaneously, measurement 2 shows that the average LOF score increases by a factor of over 5 when the discriminating feature is added. However, if one is interested in two-feature singular outliers, one may find the blue and pink line of the LOF plot interesting. See Section 4.5 for a discussion on this aspect.

4.4.5 Algae Data

The data set *Algae*, available via the UCI repository [65], comes from a water quality study where samples were taken from sites on different European rivers of a period of approximately one year. We used the (standardized) concentrations of 8 chemical substances from these samples as features. We exclude the additional measures on different algae populations. After removing rows with missing data, 306 observations and 8 features remain.

The parallel coordinates plots of Figure 4.7 show that the SODA outliers have one chemical substance that has an unusual value. One of these outliers is picked up by the LOF algorithm as well. The other two LOF outliers show unusual values for three chemical characteristics. Table 4.2 confirms the on average different nature of the outliers selected by SODA and the three conventional algorithms. If one is interested in three-feature outliers, the blue and pink lines of the LOF plot are interesting observations as well. See Section 4.5 for a discussion on this aspect.

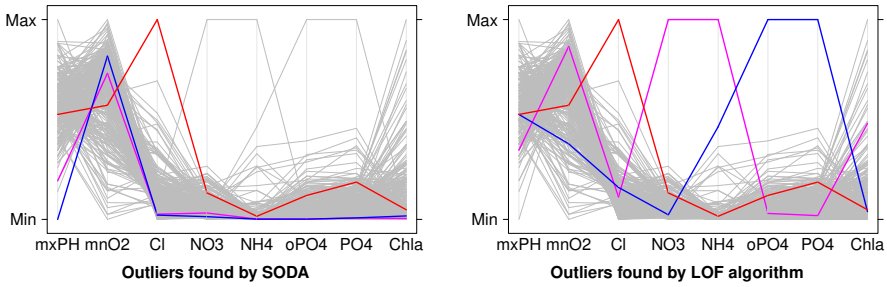


Figure 4.7: Parallel coordinates plot for the Algae Data with the three largest outliers highlighted.

4.5 Conclusion and Discussion

In this chapter, we introduced the concept of a *singular outlier* and pointed to its usefulness in the contexts of fraud detection, data quality and other areas. We introduced

an algorithm (SODA) to detect these outliers, based on the Local Euclidean Manhattan Ratio (LEMR). The algorithm has been applied to five publicly available data sets. The parallel coordinates plots, as well as the the results shown in Table 4.2, confirm that the SODA algorithm is suited for finding singular outliers.

Although Table 4.2 points out that the SODA algorithm finds observations that better match the definition of singular outliers when one restricts oneself to *one* discriminating feature, the parallel plots of the latter three data sets show that there might be interesting outliers that have two discriminating features. These observations may not get the highest outlier scores for SODA, despite the fact that the deviations in the discriminating features may be large. In such a situation, the SODA algorithm may be adapted to focus more on two-feature singular outliers or to take the absolute value of the deviations into account. This might be a fruitful aspect for further research. Part of such research can be to adjust the Local Euclidean Manhattan Ratio. The Euclidean and the Manhattan distances are based on the L_p norm. Other ratios can be constructed by taking other values for p instead of 2 and 1.

Another interesting path for further research might be to view the detection of singular outliers as an optimization problem where simultaneously attention is given to maximize dissimilarity (discriminating feature) as well as maximizing similarity (non-discriminating features).

Finally, singular outliers might be found with an algorithm with a lower time complexity than SODA. The time complexity of the SODA algorithm is dominated by finding the nearest neighbors of each observation. The nearest neighbors component gives a local flavor to SODA that might be beneficial for data sets that possess local structures (like separate clusters). However, for data sets that do not display this property, comparison of an observation $\mathbf{x}$ to the global center $\mathbf{m}_{global}$ might be used. This reduces computational time considerably.

5

Are Similar Cases Treated Similarly? A comparison between process workers

For tax administrations similar treatment of similar cases is demanded by law. Nevertheless, in practice it is not always easy to achieve this. Especially in processes involving human professional judgment (e.g., in Knowledge Intensive processes), such as audit selection, debt management and the processing of disputes, it is not even easy to verify if similar cases receive similar treatment. In these processes there is a risk of dissimilar treatment as human process workers may develop their individual experiences and convictions or change their behavior due to changes in workload or season. Awareness of dissimilar treatment of similar cases may prevent disputes, inefficiencies, or non-compliance with regulations that require similar treatment of similar cases. Therefore, in this chapter, we address the research question:

Research Question *Can we develop a data science technique that assists in testing similar treatment of similar (tax) cases?*

In this article two procedures are presented for testing in an objective (statistical) way if different groups of process workers treat similar cases in a similar way. The testing is based on splitting the event log of a process in parts corresponding to the different (groups of) process workers and analyzing the sequences of events in each part. The two procedures are demonstrated on an example using synthetic data and on a real life event log from tax data. The chapter is based on the following article:

- M. Pijnenburg and W. Kowalczyk. Are similar cases treated similarly? a comparison between process workers. In *International Conference on Business Information Systems*, pages 1–15. Springer, 2019

5.1 Introduction

Knowledge Intensive (KI) processes present an expanding topic in the field of Business Process Management (BPM), see for instance the paper of Marin et al. [75]. KI processes differ from classical processes by the presence of human process workers whose domain knowledge has an important influence on the next steps in the process. KI processes occur typically in organizations that complete complex tasks.

The human component in KI processes makes the flow of activities less predictive and raises the question whether similar cases are treated similarly. Differences in treatment of similar cases may result from differences in the expertise, workload, and opinions of the human process workers, especially when the same process is executed at two or more different sites. For example, when similar cases are presented to two process workers, they may process them in different ways.

Awareness of dissimilar treatment of similar cases by different groups of process workers (e.g., at different locations) is important for businesses, not only because a uniform treatment is usually preferred with a view on efficiency, but also since the lack of uniformity may create disputes between customers and the company. For governmental organizations similar treatment of similar cases is even more important, as similar treatment is often demanded by law or policy. As such, the topic is of interest for auditors as well. Auditors, besides the classical task of checking financial statements, are also expected to check compliance with regulations and the law.

Verifying similarity of treatments for two or more (groups of) process workers is complicated by two facts. First, the characteristics ('attributes') of cases presented to process workers may differ from process worker to process worker. For instance, when two process workers are employed in different regions, one should take into account the regional differences of the cases presented to these process workers. Second, human professionals often have the possibility (and are expected) to gather additional information about the cases they treat. Although this (unstructured) information may be stored in a case management system, it is usually not registered in the event log of the process. This 'data incompleteness' introduces an additional stochastic component when modelling the decisions of a process worker. Hence statistical techniques are required to make measure similarity of treatments, see Section 5.3 for more details.

A simple example may illustrate the issue of similar treatment. Consider a Knowledge Intensive service process of a manufacturer of laptops and printers, as shown in Figure 5.1. The service process supports customers whose device breaks down within the warranty period. A defect device that comes in, initially starts two activities that are processed in parallel: registration (V_1) and sending a confirmation (V_2). Part of the registration process is the recording of the *device type* (possible values: 'laptop' and 'printer') and the original purchase price (*price*). After registration and confirm-

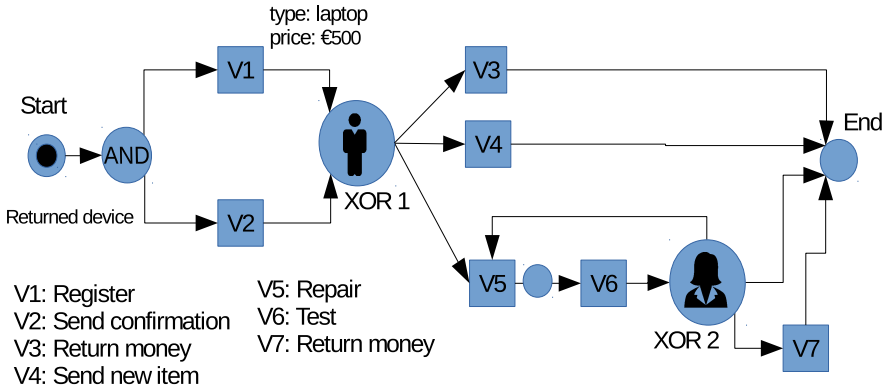


Figure 5.1: Simple process model of a service process of a laptop and printer manufacturer.

ation, a human process worker considers the defect device and has three options: return the original purchase price (V_3), send a new device (V_4), or send the defect device for repair (V_5). The choice of option depends on an assessment of the actual defect, the recorded values, and possibly additional information gathered by contacting the customer. When the repair option is chosen (V_5), the device will be send to an external repair shop and the device will be tested (V_6) afterwards. Depending on these test results, a second process worker will decide on the next activity; a good test result will lead to ship the repaired device back to the customer. Otherwise the process worker may choose to send the device again for repair or to return the money to the customer (V_7).

Traces in the event log of this process may look like this:

- `244 Printer · Start V1 V2 XOR1 V5 V6 XOR2 V5 V6 XOR2 V7 End`
- `69 Printer · Start V2 V1 XOR1 V3 End`
- `224 Laptop · Start V1 V2 XOR1 V4 End`
- `1082 Laptop · Start V1 V2 XOR1 V5 V6 XOR2 V7 End`
- `67 Printer · Start V1 V2 XOR1 V4 End`

where the first number is the purchase price of the device.

The service process is deployed at various regions. The manufacturer is interested in knowing whether similar defects receive the same treatment at each region to deal with a rise of complaints claiming that customers in region 'A' usually receive a

new item quickly, while customers in region 'B' have to wait for a repair. The service process of this manufacturer will serve as an example throughout this paper.

The research into similar treatment is motivated by two real-life examples where knowledge of similar treatment of similar cases is found to be important. One is the debt-collection process of a tax administration where it is expected that debtors in various regions are treated in a uniform way, see Section 5.4.2 for more details. The other is the treatment of patients in two wards of the same hospital according to the same protocol.

The paper is organized as follows. Section 5.2 reviews related work. Section 5.3 describes two testing procedures. Section 5.4 demonstrates these procedures on the 'warranty'-example and the tax data set. We end with conclusions and future research in Section 5.5. The code used in Section 5.4.1 can be found on github: `github.com/PijnenburgMark/Similar_Cases_Treated_Similarly/`.

5.2 Related Work

5.2.1 Process Drift

Detecting different variants of business processes is a topic that is receiving increasing attention [19], [72], [17], [82]. Most works focus on detecting changes in a business process over time, but some papers also mention differences by location (e.g., departments), like the paper of Pauwels and Calders [83], or mention the more general applicability of their method, like the paper of Bolt et al. [17].

One of the earliest papers in the field of process drift is the paper of Bose et al. [19]. The method described in this paper extracts features from an event log and compares the values of these features at two different points in time. The features demonstrated in the paper only allow for finding ' structural changes', i.e., changes in the process model.

The paper of Ostovar et al. [82] differs from the paper of Bose et al. [19] by using a technique for automated discovery of process trees and considering changes in these trees instead of using features extracted from event logs. Moreover, root cause analysis is supported by providing natural language statements to explain the change behind the drift.

In the paper of Pauwels and Calders [83] the concept of process drift is applied to the data set of the BPI challenge 2018 [108]. Besides showing concrete results, the authors add a new model-based approach relying on Dynamical Bayesian Networks. Their approach is strengthened by providing some nice aides for detecting process drift visually.

The paper of Maaradji et al. [72] adds a formal statistical test to the comparison

between different time points and thus adds objectivity. Moreover the detection of process drift is determined by looking at the frequencies of sequences of activities, and is thus capable of detecting structural changes as well as changes in frequencies of process paths.

In our approach, similarly to Maaradji et al. ([72]), we also look at the activities themselves instead of derived features and we also apply a formal statistical test. However, there are several differences. From the contextual viewpoint there are two main differences. First, we consider differences in process execution between groups of users instead of differences in time and second, we focus on the human decision making in the process and thus leaving aside the activities that are triggered automatically by the workflow management software. As a consequence of the latter, we have to consider fewer differences, making the comparison simpler and the statistical testing more powerful. From the technical point of view, there are two differences as well. First we take into account the attributes of the cases that enter the process and second, we consider the frequently occurring situation where some or all decisions in the process are independent of each other. If this assumption holds, the χ^2-test becomes much more powerful and easy to use, because of the mathematical property that the sum of independent χ^2-distributions is again a χ^2-distribution.

5.2.2 Sequence Analysis

Event logs form the basis of process mining and are in essence a number of sequences of activities. For this reason it is not surprising that techniques from sequence analysis (also known as 'sequential pattern analysis' or 'sequential pattern mining') are applicable in process mining. One of the techniques from sequence analysis that is applied frequently in this paper, is Pearson's χ^2-test to test the probabilities of going from one activity to the next against a theoretical model, see the book of Bakeman and Gottman [9]. We apply this test repeatedly under the null hypothesis of no differences in treatment between (groups of) process workers. As described by Bakeman and Gottman [9], Pearson's χ^2-test can be also applied to sequences of events, although the number of possible sequences (and thus the number of observations needed to fulfill the assumptions of the test) grows exponentially in the sequence length. We overcome this problem by making an assumption of independence of decisions (Section 5.3.1) or by focusing on the most frequent traces (Section 5.3.2).

5.3 Testing for similar treatment of similar cases

When testing similar treatment of similar cases for two groups of process workers, a first step is finding the points in the process where human process workers decide on

the next activity. We will call these points 'decision points'. In a process model these are 'XOR-junctions' (eXclusive OR-junctions), i.e., places where the flow of activities can take two or more directions.

We define 'similar treatment of similar cases' as the situation where at each decision point holds that two similar cases (measured by having the same attributes) have the same probability distribution over all possible follow-up activities.

In case only one decision point is present, the situation is relatively simple and the approach is described in Section 5.3.1. When multiple decision points are present, two situations can occur: a) the decisions at the decision points can be considered independent of each other (frequently called a 'Markov assumption' in sequence analysis). In this situation the comparing of two groups of process workers can be done for each decision point individually and the results can later be combined. This case is addresses in Section 5.3.1. b) the decisions cannot be considered independent, in other words the decision at one decision point is influenced by the previous decisions (i.e., it matters for the decision what path a case took through the process model to reach the decision point). The problem that arises in this situation is that the number of possible combinations of decisions increases exponentially in the number of decision points and hence a lot of traces in the event log are needed if a naive approach is taken. In Section 5.3.2 we present a heuristic that requires less data, by assuming that a subset of possible traces accounts for most observed traces in the event log. An assumption which is often reasonable in practice.

5.3.1 Approach 1: Independence of Decisions

5.3.1.1 A test for a single decision point

If we compare two groups of users, say from location L_1 and L_2, and we focus for the moment on *one* decision point, a table like Table 5.1 forms the basis for analysis. In the first column of Table 5.1 'Cluster of case', we see that all cases are grouped based on their attributes into K clusters. This clustering is done to make groups of similar cases. Then in the next column we see the decision that was made at the decision point, i.e., the next activity for the case. Then follow two columns indicating the count data of cases for both groups of users (L_1 and L_2). See Table 5.7 for two examples of filled out tables.

If we consider a subtable of Table 5.1 that belongs to one cluster, we can apply the well known two sample χ^2-test (see for instance the book by Kanji[56]) to test whether the differences between the two user groups can be attributed to chance, or that there is a reason to reject the hypothesis of similar treatment. If we want to apply the χ^2-test to the whole table (i.e., for all clusters simultaneously), we can take the approach as described for instance by Hald [47] (page 746). Note that Table 5.1

		group of process workers	
Cluster of case	output activity	L_1	L_2
x = 1	V^1	$n_1^{1,1}$	$n_2^{1,1}$
	$\vdots$	$\vdots$	$\vdots$
	V^q	$n_1^{1,q}$	$n_2^{1,q}$
x = 2	V^1	$n_1^{2,1}$	$n_2^{2,1}$
	$\vdots$	$\vdots$	$\vdots$
	V^q	$n_1^{2,q}$	$n_2^{2,q}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
x = K	V^1	$n_1^{K,1}$	$n_2^{K,1}$
	$\vdots$	$\vdots$	$\vdots$
	V^q	$n_1^{K,q}$	$n_2^{K,q}$

Table 5.1: Three-way contingency table needed to apply the χ^2-test for one decision point for two groups of process workers.

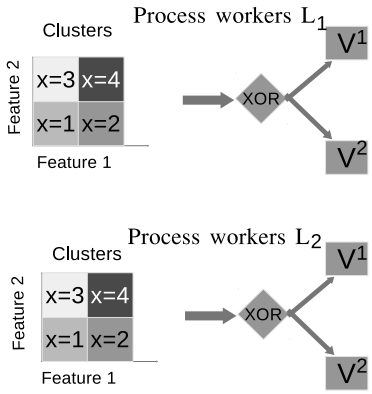


Figure 5.2: Figure illustrating the elements of the table at the left

contains only two locations. This is only for demonstrative purposes as the χ^2-test works equally well for any number of user groups.

Mathematically the test for one decision point comes down to the following. To test the null hypothesis of similar treatment of similar cases, we apply the (two sample) χ^2-test. So we calculate,

$$\chi^2_{XORj} = \sum_{i \in \text{all cells}} \frac{(O_i - E_i)^2}{E_i} = \sum_{x,V,L} \frac{(n_L^{x,V} - E_L^{x,V})^2}{E_L^{x,V}}, \tag{5.1}$$

where $E_L^{x,V}$ denotes the expected number of counts. These expected numbers are estimated assuming no differences between the groups of process workers ($\mathcal{H}_0$), so,

$$E_L^{x,V} = \frac{n_\bullet^{x,V} n_L^{x\bullet}}{n_\bullet^{x\bullet}}, \tag{5.2}$$

where,

$$n_\bullet^{x,V} = \sum_L n_L^{x,V}, \qquad n_L^{x,\bullet} = \sum_V n_L^{x,V}, \qquad n_\bullet^{x\bullet} = \sum_{V,L} n_L^{x,V}. \tag{5.3}$$

For one cluster (e.g., $x = 1$), the degrees of freedom are equal to $(q-1)(t-1)$ [47] (where t is the number of groups of process workers, and q the number of next activities). Since there are K clusters, we have,

$$df_{XORj} = K(q-1)(t-1). \tag{5.4}$$

The statistic (5.1) is χ^2-distributed with degrees of freedom given in (5.4) if the assumptions of the Pearson's χ^2-test are fulfilled.

Experiments have demonstrated that the assumptions of Pearson's χ^2-test are sufficiently fulfilled if *no more than 20% of the expected counts are less than 5 and all individual expected counts are 1 or greater*, see the book by Yates et al. [115]. One way of meeting this condition is to find a balance in the number of clusters K, such that K is small enough for each cluster to contain enough cases, while simultaneously keeping K large enough to make sure that cases that are considered as dissimilar by experts are in different clusters. See Section 5.4 for the values used in the experiments. If after choosing an appropriate number of clusters the assumptions of Pearson's test are still violated, we propose to remove rows from Table 5.1 with infrequent row sums (i.e., activities that are seldom chosen by any groups of process workers) and apply equation (5.1) on the reduced table. Since the frequency of these activities is low for all process workers, the effect of removing will be small in practice. Of course, the degrees of freedom have to be adjusted. If l rows are omitted, then the number of degrees of freedom is given by:

$$df^{adj}_{XORj} = K(q-1)(t-1) - l(t-1). \tag{5.5}$$

By introducing two groups of process workers L_1 and L_2 and comparing the model of a decision point for these two groups, we must be cautious that possible differences are only caused by differences in treatment of the groups of process workers L_1 and L_2 and *not* by differences in the attributes of the cases that are presented to group L_1 and group L_2. Otherwise we would erroneously conclude there is dissimilar treatment of similar cases, while in reality there is dissimilar treatment of *dis*similar cases. In part, these differences in the cases of each group are prevented by the clustering the cases into K clusters based on the known attributes. However, typically not all attributes are recorded. If one suspects the existence of an attribute that has an influence on the decision taken by the process workers and this feature is also related to the group of users a case is assigned to, one must, within each cluster of cases, distribute cases randomly over the groups of process workers. This way one ensures the same probability distribution for the groups and one can safely apply the tests.

5.3.1.2 Multiple decision points that are independent

If there are multiple decision points in the process and we can assume that the decisions by the human process workers are independent from previous decisions in the process, we can easily extend our χ^2-test. When the statistics χ^2_{XORi} have been computed for the individual decision points, a test of similar treatment for the whole process (i.e., all m decision points) can be constructed by adding the individual statistics as well as the degrees of freedom:

$$\chi^2_{overall} = \chi^2_{XOR1} + \ldots + \chi^2_{XORm}, \tag{5.6}$$

with

$$df_{overall} = df^{adj}_{XOR1} + \ldots + df^{adj}_{XORm}. \tag{5.7}$$

The final statistic $\chi^2_{overall}$ is χ^2-distributed with $df_{overall}$ degrees of freedom since the sum of independent χ^2-statistics is again χ^2-distributed with the degrees of freedom equal to the sum of the original ones. The independence of the individual statistics is ensured by the Markov assumption.

The value of the statistic $\chi^2_{overall}$ leads to a p-value indicating the probability that the hypothesis $\mathcal{H}_0$ is due to the randomness in the sample. A value lower than 0.05 is generally accepted as a rejection of $\mathcal{H}_0$ and is thus evident of dissimilar treatment of similar cases.

The complete test procedure in case of independent decision points is summarized in Test Procedure 1.

Test Procedure 1: Procedure to test on similar treatment by two or more groups of process workers when decision points are independent

Input : Event logs
Output: A p-value
1 Find the decision points in the process;
2 **for** *each decision point* **do**
3 Find relevant attributes;
4 Cluster the cases of all event logs into K clusters based on the attributes (perform the clustering on the collective data of all user groups);
5 Construct a contingency table like Table 5.1;
6 Check the assumptions of Pearson's χ^2-test and possibly remove rows with low row sums (adjust degrees of freedom accordingly);
7 Use equation (5.1) to compute the test statistic for this decision point, and equation (5.5) for the degrees of freedom;
8 **end**
9 Apply equation (5.6) and (5.7) and read off the p-value from standard tables of the χ^2-distribution.

5.3.2 Approach 2: Frequent Paths

For some processes the Markov assumption will not hold. A straightforward generalization in this case is to cluster cases and consider for each cluster all possible traces and test if some traces occur significantly more frequently for one group of process workers. Although this approach is in line with the approach taken in the previous section, it has the drawback that a lot of data is needed in order to satisfy the underlying assumptions of Pearson's χ^2-test. Namely, the number of possible traces grows exponentially in the number of decision points. For this reason we will work out an approach that does not take into account *all* traces, but only the most frequent ones (i.e., 'typical paths') for each cluster of similar cases. Then the observed frequencies of these traces for each group of process workers are compared with the expected frequencies (based on the average frequencies over all process workers) using the χ^2-test.

Technically, the frequent paths approach comes down to first clustering cases based on known features, and then finding the most frequent traces for each cluster. If the event log contains many traces, special algorithms can be applied to find the most frequent traces, like the SPADE algorithm described by Zaki [116].

Subsequently, a table like Table 5.2 is created that contains the frequencies of these frequent paths for all groups of process workers. Then equation (5.8) can be

		group of process workers	
Cluster of case	frequent trace	L_1	L_2
x = 1	$V^1V^3V^5$	$n_1^{1,1}$	$n_2^{1,1}$
	⋮	⋮	⋮
	$V^2V^3V^4$	$n_1^{1,r}$	$n_2^{1,r}$
x = 2	⋮	⋮	⋮
⋮	⋮	⋮	⋮
x = K	V^1V^5	$n_1^{K,1}$	$n_2^{K,1}$
	⋮	⋮	⋮
	$V^1V^2V^3V^5$	$n_1^{K,s}$	$n_2^{K,s}$

Table 5.2: Three-way contingency table needed to apply the χ^2-test for all decision points for two groups of process workers when the assumption of independence of decision points does not hold.

applied to each frequent trace s in cluster x,

$$\chi^2 = \sum_{i \in \text{all cells}} \frac{(O_i - E_i)^2}{E_i} = \sum_{x,s,L} \frac{(n_L^{x,s} - E_L^{x,s})^2}{E_L^{x,s}}, \tag{5.8}$$

where $E_L^{x,s}$ is estimated by,

$$E_L^{x,s} = \frac{n_\bullet^{x,s} n_L^{x\bullet}}{n_\bullet^{x\bullet}}, \tag{5.9}$$

and,

$$n_\bullet^{x,s} = \sum_L n_L^{x,s}, \qquad n_L^{x,\bullet} = \sum_s n_L^{x,s}, \qquad n_\bullet^{x\bullet} = \sum_{s,L} n_L^{x,s}. \tag{5.10}$$

The degrees of freedom of the statistic in equation (5.8) equals:

$$df = (N - K)(t - 1), \tag{5.11}$$

where N is the number of rows in the contingency table (Table 5.2), K the number of clusters and t the number of groups of process workers. This can be seen by realizing that for each cluster i the degrees of freedom equals $(s_i - 1)(t - 1)$ [47], where s_i is the number of frequent paths for cases in cluster i. The Test Procedure for the Frequent Path approach is presented in Test Procedure 2.

Test Procedure 2: Procedure to test on similar treatment by two or more groups of process workers when decision points are dependent and most traces belong to a few frequently occurring traces

Input : Event log
Output: A p-value

1 Cluster all cases based on their features into K clusters;
2 Determine for each cluster the most frequent traces in the event log, for instance by applying a frequent sequence algorithm like SPADE. Discard all other traces;
3 Make a contingency table like Table 5.2;
4 Use equation (5.8) to compute the test statistic, and equation (5.11) for the degrees of freedom;
5 Find the p-value from standard tables of the χ^2-distribution.

		XOR-junctions	
		behavior I	behavior II
Features	distribution I	event log 1	event log 2
	distribution II	event log 3	event log 4

Table 5.3: Differences of the four generated event logs

5.4 Experiments

5.4.1 Synthetic data

In this section we will apply the test procedures to the example mentioned in the introduction. The code used for this can be found on github: `github.com/PijnenburgMark/Similar_Cases_Treated_Similarly/`.

We used the process model of Figure 5.1 to generate four event logs. Each event log consists of 500 traces and the value of two meaningful features for each case (price and type). The four event logs differ in the behavior of the two decision points and in the distribution of the two features, see Table 5.3.

Table 5.6 specifies the behavior I and behavior II of the decision points as well as the distributions of the input cases. For instance we see that under behavior I, a printer that is returned with an original purchase price of 200 has the probability of 0.2 for being send for repair, while this probability under behavior II is 0.6. Moreover we see that the behavior of decision point 2 (XOR 2) depends on the number of times

		2^{nd} Event Log			
		event log 1	event log 2	event log 3	event log 4
1^{st} Event Log	event log 1	1.000000	0.000000	0.975989	0.000000
	event log 2	0.000000	1.000000	0.000000	0.057141
	event log 3	0.975989	0.000000	1.000000	0.000000
	event log 4	0.000000	0.057141	0.000000	1.000000

Table 5.4: p-values resulting from applying Test Procedure 1 to all pairs of the four event logs. We used 10 clusters.

the device has been send for repair before (with two options: one time or more than one time) as well as the result of the test that is performed after the repair (activity V_6 in Figure 5.1). Note that if the results of the test are positive, the device is ready and thus there is a probability of one that the process trace will be ended. In our example we set the probability that the test of V_6 gives a positive result to 0.7.

Clearly, event logs 1 and 3 have the same process (i.e., no dissimilar treatment), but differ in the input (event log 1 mainly cheap printers, event log 3 more laptops). In contrast event logs 2 and 4 are generated with other process parameters. In these latter two event logs the process workers have a preference for the repair option instead of sending a new item or returning the money.

We will compare the four event logs pairwise. Large p-values (over 0.05) are expected when comparing event logs that have the same behavior (i.e., event log 1 with event log 3, and event log 2 with event log 4), while low p-value are expected if we compare event logs with different behavior such as event log 1 and event log 2.

The p-values resulting of applying Test Procedure 1 are shown in Table 5.4. The application of this test procedure is justified because the decision points are independent. The results are as expected, i.e., the test procedure is able to clearly distinguish dissimilar treatment from differences in the input. For feature selection we used a standard feature selection method from python's sci-kit learn package based on the ANOVA F-value, and for the clustering we applied a standard Gaussian Mixture clustering algorithm from the same package. The number of clusters was chosen $K = \#$ cases $/100$, where '# cases' are the number of cases that go through the decision point in the event log (for both groups).

We also applied Test Procedure 2, resulting in the p-values of Table 5.5. These values demonstrate that our algorithm properly captures relevant properties of this synthetic data set. In applying Test Procedure 2 we used again Gaussian Mixture clustering. The number of clusters was chosen $K = \#$ traces $/250$, where '# traces' are the number of traces of the combined event logs for both group of process workers.

		2^{nd} Event Log			
		event log 1	event log 2	event log 3	event log 4
1^{st} Event Log	event log 1	1.000000	0.000000	0.982920	0.000000
	event log 2	0.000000	1.000000	0.000000	0.662541
	event log 3	0.982920	0.000000	1.000000	0.000000
	event log 4	0.000000	0.662541	0.000000	1.000000

Table 5.5: p-values resulting from applying Test Procedure 2 to all pairs of the four event logs. We used 4 clusters.

5.4.2 Tax Debt Collection data

For demonstration purposes, we applied Test Procedure 1 to real data of the Netherlands Tax and Customs Administration (NTCA). In the year 2013 a debt collection process was in place, a part of which is sketched in Figure 5.3. As soon as a debt is

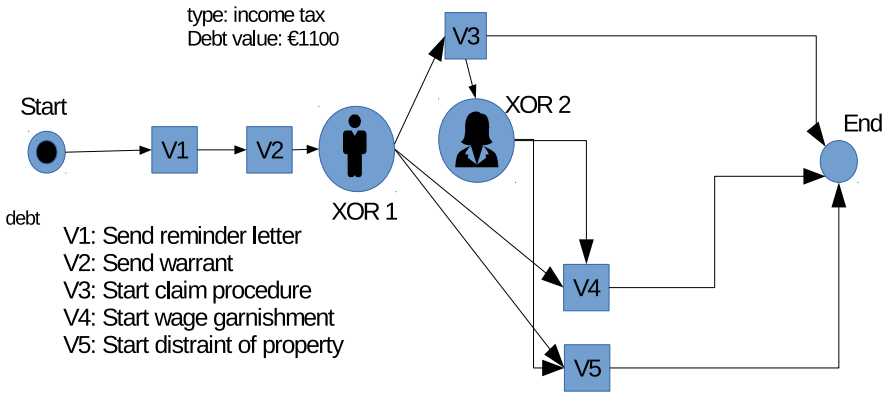


Figure 5.3: Part of the debt collection process of the NTCA in 2013.

over its due date, automatically a reminder letter is sent. When no payment has taken place, a legal notice (warrant) is sent that allows for legal actions afterwards. Usually, there are several legal actions possible and the choice is determined by a human process worker (XOR 1). The legal actions in this part of the process are: a special claim procedure that allows to take money of a savings account (V_3), wage garnishment (V_4), and distraint of property, e.g., a car (V_5). When the special claim procedure V_3 does not lead to payment of the debt, a second human process worker (XOR 2) can decide for actions V_4 or V_5.

We have compared event logs from two locations. We took 1000 debts from each location in the year 2013. We took into account two features of each debt: the debt

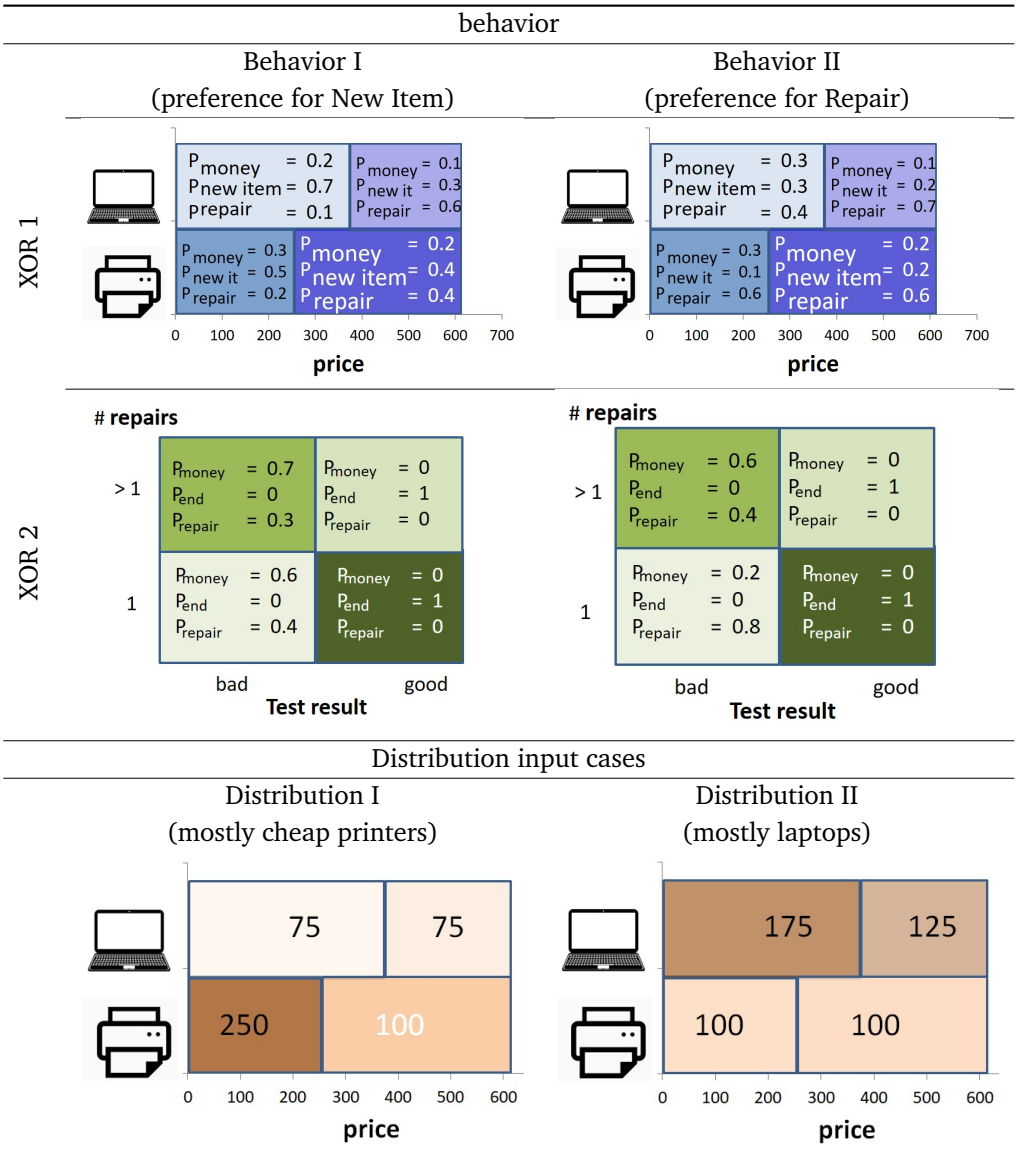


Table 5.6: Worked out example: two types of behavior of the XOR junctions and two distributions of the input cases.

value and the tax type and restricted ourselves to two tax types. The debt value has been used for constructing the clusters for both decision points, while the tax type played a role for the clustering of XOR 1 only. The contingency tables for both decision points are displayed in Table 5.7.

XOR 1		location	
Cluster	Activity	A	B
1	V_3	16	14
1	V_4	42	48
1	V_5	21	17
2	V_3	1	2
2	V_4	16	39
2	V_5	64	35
3	V_3	381	379
3	V_4	25	59
3	V_5	80	90
4	V_3	217	213
4	V_4	8	47
4	V_5	59	50
	Total	930	993

XOR 2	location		
Cluster	Activity	A	B
1	V_4	17	17
1	V_5	64	53
2	V_4	8	9
2	V_5	42	24
	Total	131	103

Table 5.7: Contingency tables belonging to the decision points XOR 1 and XOR 2 of the tax collection data. Note we were not able to analyze 77 of the 2000 cases.

The value of the χ^2-statistic for the first decision point is 59.247 (8 degrees of freedom), while 1.785 (2 degrees of freedom) for the second decision point. For the combination of the two decision point we find, under the assumption of independent decisions, a value of 61.032 (10 degrees of freedom) which corresponds to a p-value of $2.31 \cdot 10^{-9}$. The low p-value indicates that, under this model, we should reject the hypothesis of similar treatment of similar cases. Due to privacy reasons we did not use some important features that could lead to different results.

5.5 Conclusion and Future Research

The paper started by noting that the human factor in Knowledge Intensive processes raises the question of similar treatment by several groups of process workers. The two test procedures that are proposed in Section 5.3 are able to test similarly treatment as is demonstrated in Section 5.4 on an artificial example of a service process of a hardware manufacturer and a debt collection process of a tax administration.

Tax administrations may employ these tests to see in what processes the null hypothesis of similar treatment of similar cases does not hold. The tests allow to pinpoint at what step in the process this occurs and measures may be taken to restore the similar treatment.

The two test procedures presented in this paper are based on rather element-

ary statistical techniques. We choose deliberately to solve the problem with elementary techniques as simple methods display the nature of the problem most clearly. Moreover an elementary approach allows to communicate clearly with business experts. Besides an elementary approach allows for easy extensions to meet more particular needs. For instance most organizations may tolerate a certain small level of dissimilar treatment and the test may be extended to take this into account. However, applying some recently developed algorithms may have some advantages as well. In particular recurrent neural networks can be used successfully for modelling complex sequential data [68]. Unfortunately these networks are 'black-box models' and provide little insight into the nature of detected differences in the behaviour of groups of process workers.

Finally note that other interpretations of 'similar treatment' are possible and might be appropriate in some settings. In this paper similar treatment has been defined in terms of probabilities of going to the next activity in the process model. However similar treatment can also be defined for instance in terms of the amount of time that is spend on each case, or as the level of expertise that is involved in each case. Tests based on these metrics have not been explored yet, but may be the subject of further research.

6

Extending an Anomaly Detection Benchmark with Auto-encoders, Isolation Forests, and RBMs

In tax administrations and other organizations that apply data science techniques, a frequently heard question is *What algorithm should I use?* In this chapter we make a contribution to answering this question by addressing a class of techniques, *unsupervised anomaly detection algorithms*, that are frequently used in tax administrations. The exact research question is:

> **Research Question** *Unsupervised anomaly detection algorithms play an important role in (tax) fraud detection. What algorithms can be expected to work well under what conditions?*

In this chapter, the recently published benchmark of Goldstein and Uchida [43] for unsupervised anomaly detection is extended with three anomaly detection techniques: Sparse Auto-Encoders, Isolation Forests, and Restricted Boltzmann Machines. The underlying mechanisms of these algorithms differ substantially from the more traditional anomaly detection algorithms, currently present in the benchmark. Results show that in three of the ten data sets, the new algorithms surpass the present collection of 19 algorithms. Moreover, a relation is noted between the nature of the outliers in a data set and the performance of specific (clusters of) anomaly detection algorithms. The chapter is based on the following article:

- M. Pijnenburg and W. Kowalczyk. Extending an anomaly detection benchmark with auto-encoders, isolation forests, and rbms. In *International Conference on Information and Software Technologies*. Springer, 2019. Best Paper Award

6.1 Introduction

The expanding number of anomaly detection algorithms creates the need to compare algorithms objectively. Goldstein and Uchida [43] made a start by providing a benchmark for unsupervised anomaly detection, consisting of 10 data sets and 19 algorithms. In this chapter, we extend the number of algorithms by adding three anomaly detection algorithms: Sparse Auto-encoders, Isolation Forests, and Restricted Boltzmann Machines (RBMs).

The three algorithms are interesting since they have a different underlying mechanism compared to the current algorithms in the benchmark: two of the new algorithms, Sparse Auto-encoders and RBMs, originate from the popular field of (deep) neural networks. Within this field, they are among the simplest and best-known algorithms for detecting anomalies [60]. The third algorithm is based on random trees. The underlying mechanisms differ substantially from the more traditional algorithms in the benchmark that are mostly distance-based, like the k-Nearest Neighbors algorithm and the Local Outlier Factor. Moreover, when auto-encoders and RBMs are applied to anomaly detection, it is usually on image data [103], [102] or sequential data [73], [114]. Hence it is of interest to see their performance on the (small) classical tabular data used in the benchmark [43].

The chapter is organized as follows. In Section 6.2, the three anomaly detection algorithms are described. Then, in Section 6.3, the actual experiments performed are described as well as the data sets of the benchmark. In Section 6.4 we present the results of the experiments and compare these with results mentioned in [43]. The chapter ends with Conclusions and Discussion in Section 6.5. The code used in the experiments is published at [86].

6.2 Theoretical Background

6.2.1 Sparse Auto-encoder

Standard auto-encoders are neural networks with architecture as depicted in Figure 6.1. Sometimes it is required that $W_1 = W_2^t$, for regularization purposes. We will not impose this restriction in this chapter. In its most basic form, as shown in the figure, the network consists of one input layer, a hidden layer, and an output layer with the same number of nodes as the input layer. The network is trained by providing the same observation $\mathbf{v}$ as input *and* output, and requiring to minimize the reconstruction error $\|\text{auto}(\mathbf{v}) - \mathbf{v}\|^2$. Essentially, the network must learn the identity function ('auto' means 'self' in Greek).

The number of nodes in the hidden layer is intentionally limited, such that the en-

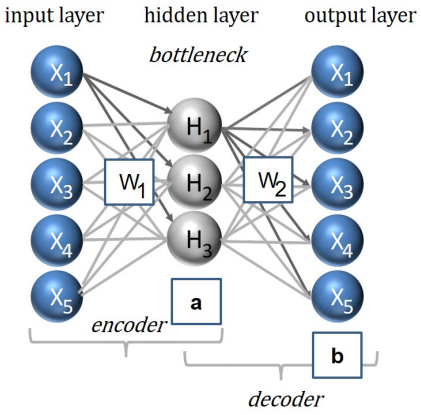


Figure 6.1: Architecture of the simplest form of a standard auto-encoder with one hidden layer. In general, the encoder and the decoder part may consist of more complex neural networks.

coder has to extract the essential information from the input features in order for the decoder to reconstruct the original input as closely as possible. The encoder part of an auto-encoder can thus be seen as a dimensionality-reducing algorithm, reducing the original dimensionality of the input space to the dimensionality of the space formed by the hidden nodes.

Training an auto-encoder is usually done by gradient descent, in particular *stochastic* gradient descent. The latter algorithm speeds up convergence in comparison with standard gradient descent and also introduces some noise that helps to avoid local minima. Backpropagation is generally used to compute the gradient by going backward from the output layer to the input layer.

In the experiments, see Algorithm 1, we used the BFGS (Broyden Fletcher Goldfarb Shanno) algorithm, a quasi-Newton optimization algorithm for minimizing the target function. The BFGS algorithm works well for data sets with a small number of features as are most data sets in the benchmark. The target function in the experiments consists of the reconstruction error – equation (1) in Algorithm 1 –, a standard L_2 regularization term (2) that is added more routinely in training neural networks nowadays, and a sparsity constraint term (3) that is typical for *sparse* auto-encoders.

Sparse auto-encoders are a variant on classical auto-encoders where the 'bottleneck' is not created by limiting the number of hidden nodes, but by requiring that only a limited number of hidden nodes have a high activation value for each observation.

When applying auto-encoders to anomaly detection, at least two approaches may be taken. The first approach uses the dimensionality-reduction characteristic of the

Algorithm 1: Training and Scoring of an auto-encoder with one hidden layer.

Input : A data set $\mathbf{V}$ with p (numeric) features and n observations. We assume each column to be scaled into the range [0,1] using min-max scaling.
$h = \{5, 10, \lfloor p/2 \rfloor, p\}$ the number of hidden nodes,
$\lambda = 0.001$ the learning rate,
$\rho = 0.1$ sparsity hyper-parameter,
$\beta = 0.05$ factor influencing the relative importance of the sparsity term in the cost function,

Output: Reconstruction error of all observations $\mathbf{v} \in \mathbf{V}$.

1 Initialize weight matrix W with random number from a $\mathcal{N}(0, 0.01)$ distribution

2 Call a standard BFGS optimizer for minimizing the target function that consists of the reconstruction error, a regularization term and the sparsity constraint:

3

$$J(W_1, W_2, \mathbf{a}, \mathbf{b}) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\|\text{auto}(\mathbf{v}_i) - \mathbf{v}_i\|^2 \quad (6.1)$$

$$+\frac{\lambda}{2}(\|W_1\|^2 + \|W_2\|^2) \quad (6.2)$$

$$+\beta\sum_{i=1}^{n}[\rho\log\frac{\rho}{\widehat{\rho}(\mathbf{v}_i)} + (1-\rho)\log\frac{1-\rho}{1-\widehat{\rho}(\mathbf{v}_i)}], \quad (6.3)$$

where auto($\mathbf{v}$) is the output of the feedforward pass through the network:

$$\text{auto}(\mathbf{v}) = 1/(1+\exp(-W_2 A(\mathbf{v}) - \mathbf{b})), \quad (6.4)$$

$$A(\mathbf{v}) = 1/(1+\exp(-W_1\mathbf{v} - \mathbf{a})), \quad (6.5)$$

and $\widehat{\rho}(\mathbf{v}))$ is the average activation of the nodes in the hidden layer:

$$\widehat{\rho}(\mathbf{v})) = \frac{1}{h}\sum_{i=1}^{h} A_i(\mathbf{v}). \quad (6.6)$$

4 After convergence, pass all observations $\mathbf{v} \in \mathbf{V}$ through the trained auto-encoder and compute the reconstruction error $e(\mathbf{v}) = \|\text{auto}(\mathbf{v}_i) - \mathbf{v}_i\|^2$.

5 **return** the vector of reconstruction errors: $e(\mathbf{v}_1), \ldots, e(\mathbf{v}_n)$.

auto-encoder part. Once trained, the auto-encoder will transform the original features into a new low-dimensional feature space (i.e., the hidden layer). In the new feature space, traditional distance-based anomaly detection techniques may be applied that would not work properly in the original, high-dimensional space due to the 'curse of dimensionality'. This first approach works in particular well for standard auto-encoders. The second approach will work for standard auto-encoders as well as sparse auto-encoders. The idea underlying this approach is that the network will only achieve a low reconstruction error if it focuses on frequently occurring patterns. As a result, observations belonging to infrequent patterns will receive a high reconstruction error. Hence the reconstruction error can serve as an anomaly score. This approach is adopted in this chapter.

6.2.2 Isolation Forest

Isolation forest [71], see Algorithm 5, is an algorithm specifically developed to find anomalies. It resembles the random forest algorithm. As such it is a collection of many trees. However these trees are no decision trees, but 'random trees'. A 'random tree' is a tree where each split involves a randomly selected feature, which is split based on a random value. The underlying assumption of an isolation forest is that anomalies are few and have different values from most observations. As a result, anomalies will often be isolated from the other observations in very few splits. Therefore, by observing the leaf of an observation in many trees, and computing the average distance of these leaves to the root of the trees, anomalies will have a small distance, while normal observations will have a large distance. Hence, the average distance to the root can be used as an anomaly score.

6.2.3 Restricted Boltzmann Machine

A Restricted Boltzmann Machine (RBM) is a stochastic neural network with two layers: a *visible layer*, and a *hidden* layer, see Figure 6.2. The network has no output layer like auto-encoders, and signals in the network travel back and forth between the two layers, starting at the visible layer. Both layers are fully connected, i.e., each input node is connected to each hidden node, and vice versa. Moreover the weights of the connections are symmetric $W_{ij} = W_{ji}$. No connections between nodes of the same layer are allowed. All nodes in a *classical* RBM are binary, i.e. can take two values: 0 and 1. This in contrast to auto-encoders. Consequently, numerical inputs have to be discretized and transformed to binary dummy variables. In our experiments, the 'thermometer encoding' is used for the latter, as it preserves the ordering present in numerical features. The difference between thermometer encoding and standard

Algorithm 2: Training and Scoring of Isolation Forest

Input : A data set $\mathbf{V}$ with p (numeric) features. We assume each column to be scaled into the range [0,1] using min-max scaling.
$n\_tree = 100$ the number of trees,
$hlim = 8$ the maximum depth of a single tree,
$n\_samp = 256$ number of observations used in the training sample,
$min\_leaf = 1$ minimum number of observations in a leaf.

Output: Scaled Version of the average path length of each observation $\mathbf{v} \in \mathbf{V}$.

```
Create n_tree random trees:
for i in 1 ... n_tree do
    Take a random sample X ⊂ V of size n_samp
    Initialize first node of tree
    while there is a node N with depth < hlim and # observations > min_leaf do
        randomly select an attribute q
        randomly select a split point s ∈ [0, 1]
        Split node N in 2: q <= p, q > p
    end
end
Compute (scaled version of) Average Path Length, APL(v), for all observations v ∈ V
return the vector of average path lengths: (APL(v_1), ..., APL(v_n)).
```

dummy encode can best be illustrated by the example of representing the value 0.45 of continuous feature with a range $[0, 1]$ that is discretized in ten equal width bins $[0, 0.1), [0.1, 0.2), \ldots$. With standard encoding 0.45 falls in the bin $[0.4, 0.5)]$ and will be representated by the vector $(0, 0, 0, 0, 1, 0, 0, 0, 0, 0)$. With thermometer encoding 0.45 would be represented by the vector $(1, 1, 1, 1, 1, 0, 0, 0, 0, 0)$.

An RBM can be interpreted as a graphical model, or a 'Markov Random Field', see [39]. Consequently, there is a model for the probability distribution over the feature space:

$$p(\mathbf{v}) = \frac{1}{Z} \sum_{\text{all feasible } \mathbf{h}} e^{-E(\mathbf{v},\mathbf{h})}, \tag{6.7}$$

where Z is a normalization constant (also known as the *partition function*) ensuring that $\sum p(\mathbf{v}) = 1$,

$$Z = \sum_{\substack{\text{all feasible} \\ \mathbf{v},\mathbf{h}}} e^{-E(\mathbf{v},\mathbf{h})}, \tag{6.8}$$

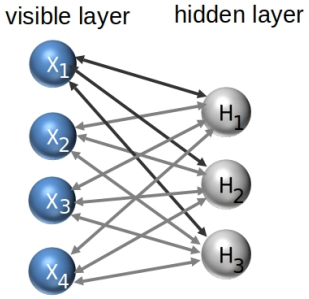


Figure 6.2: Architecture of a Restricted Boltzmann Machine.

and E is the so-called *energy function*,

$$E(\mathbf{v}, \mathbf{h}) = -\sum_{i \in \text{visible}} a_i v_i - \sum_{j \in \text{hidden}} b_j h_j - \sum_{i,j} w_{ij} v_i h_j. \tag{6.9}$$

This family of probability distributions is known as 'Boltzmann distributions' and has been subject of study in statistical physics.

Training an RBM amounts to adjusting the parameters of the energy function (6.9) such that the distribution fits the observations of the training data set, see Algorithm 6. Fitting the distribution to observations is done by maximizing the likelihood over the the training set V,

$$\arg\max_{a_i, b_i, w_{ij}} \prod_{\mathbf{v} \in V} p(\mathbf{v}). \tag{6.10}$$

The maximization is usually done by with the help of Contrastive Divergence, see [39], a 'stochastic gradient descent'-like algorithm, showing fast convergence at the cost of approximating the gradient.

After training, the probability of each observation $p(\mathbf{v})$ may be computed by equation (6.7). In practice, however, applying equation (6.7) requires computing the partition function (6.8), which is computationally intractable. Instead, one may notice that $p(\mathbf{v})$ is proportional to,

$$p(\mathbf{v}) \propto e^{-F(\mathbf{v})} = \sum_{\mathbf{h}} e^{-E(\mathbf{v}, \mathbf{h})}. \tag{6.11}$$

Here $F(\mathbf{v})$ is the *free energy of an observation* $\mathbf{v}$: the energy that a single configuration would need to have in order to have the same probability as all of the configurations that contain $\mathbf{v}$ [51].

The free energy of an observation can be calculated in linear time, due to the special architecture of a *Restricted* Boltzmann Machine, which does not allow links between hidden nodes, leading to an energy function (6.9) that involves no cross

Algorithm 3: Training and Scoring of an RBM using CD-1

Input : A data set $\mathbf{V}$ with p numeric features, scaled into the range [0,1]
$h = \{5, 10, \lfloor p/2 \rfloor, p\}$ the number of hidden nodes,
$b = \{3, 5, 7, 10\}$ the number of bins for each feature,
$k = 10$ batch size,
$\lambda = 0.01$ initial learning rate,
$m = 0.95$ momentum term (only used for bias vectors)

Output: for each $\mathbf{v} \in \mathbf{V}$: $\exp(-F(\mathbf{v}))$. This expression is proportional to $p(\mathbf{v})$.

Discretize columns of $\mathbf{V}$ into b bins each, using equal width binning followed by thermometer encoding. Denote the new data set with $p \cdot b$ binary columns $\mathbf{V}'$.
Initialize values of matrix W and vectors $\mathbf{a}$ and $\mathbf{b}$ with small uniform random numbers
while *convergence = FALSE* **do**
  randomize order of rows and put rows in batches of size k
  for *each batch* V_0 **do**
    sample binary values H_0 based on values V_0:
    $H_0 = \text{random}_{\{0,1\}}(1/(1 + \exp(-WV_0^t - \mathbf{b})))$
    compute new values V_1 based on H_0: $V_1 = 1/(1 + \exp(-(H_0W)^t - \mathbf{a}))$
    Compute probabilities of hidden nodes H_1 based on V_1 (without sampling): $H_1 = 1/(1 + \exp(-WV_1^t - \mathbf{b}))$
    Adjust weights:
    $W = W + \lambda \cdot (H_0^t V_0 - H_1^t V_1)/k$
    $d\mathbf{a} = d\mathbf{a}_{-1} \cdot m + \lambda \cdot$ column_means $(V_0 - V_1)$
    $\mathbf{a} = \mathbf{a} + d\mathbf{a}$
    $d\mathbf{b} = d\mathbf{b}_{-1} \cdot m + \lambda \cdot$ column_means $(H_0 - H_1)$
    $\mathbf{b} = \mathbf{b} + d\mathbf{b}$
  end
  compute the total free energy $F(\mathbf{V}') = \sum_{\mathbf{v} \in \mathbf{V}'} F(\mathbf{v})$, see equation (6.12)
  Check on convergence:
  if $|(F(\mathbf{V}') - F_{previous}(\mathbf{V}'))/F_{previous}(\mathbf{V}')| < 0.001$ **then**
    convergence = TRUE
  end
  Check on adjustment of λ:
  if $(F(\mathbf{V}') - F_{previous}(\mathbf{V}'))/F_{previous}(\mathbf{V}') > 0.01$ **then**
    $\lambda = 0.1 \cdot \lambda$
  end
end
return for each $\mathbf{v} \in \mathbf{V}'$: $\exp(-F(\mathbf{v}))$, see equation (6.12).

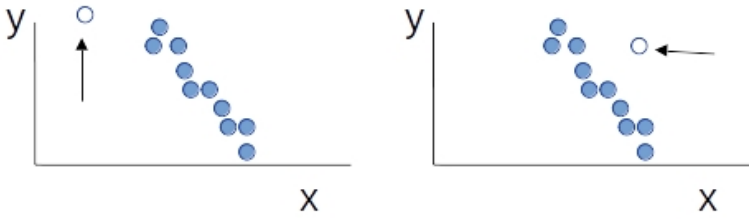


Figure 6.3: 'Univariate' outlier (left, U-index = 0.74) and 'multivariate' outlier (right, U-index = 0.43).

terms like $h_j h_k$. Hence the sum over all possible values of $\mathbf{h}$ in equation (6.11), comes down to summing over each element of $\mathbf{h}$ separately, essentially reducing the time to compute the free energy of an observation from exponential to linear. The free energy of an observation can be expressed most conveniently as,

$$F(\mathbf{v}) = -\sum_{i \in \text{visible}} v_i a_i - \sum_{j \in \text{hidden}} \log(1 + e^{x_j}), \tag{6.12}$$

where $x_j = b_j + \sum_i v_i w_{ij}$ is the input for the hidden node j, see equation (22) of [39] for a derivation.

Applying the procedure above, we obtain for each observation the expression $\exp(-F(\mathbf{v}))$, which is proportional to $p(\mathbf{v})$. Now one may proceed along two lines: (1) if interest lies only in obtaining the k most anomalous cases in a data set, one may order the observations according to $F(\mathbf{v})$ and report the k observations with largest free energy. (2) otherwise, one may obtain a set of outliers by computing the median m and interquartile range IQR of $\exp(-F(\mathbf{v}))$ for all $\mathbf{v} \in V$ and set a threshold $\theta = m - c \cdot IQR$ below which observations are considered outliers.

6.2.4 Type of outliers

A distinction between type of outliers, that will prove fruitful in explaining differences among algorithms in Section 6.4, is that some outliers are multivariate in nature while others are univariate, compare Figure 6.3. In the left subplot a univariate outlier is shown. This outlier can be detected by only looking at one feature (x), while the multivariate outlier of the right subplot requires knowledge about both features.

In practice we do not know a priori the nature of the outliers, as we do not know what observations are outliers. However, in a benchmark situation, we *know* what observations are outliers. The outliers are indicated by a binary feature y taking the value 1 for an outlier and 0 otherwise. In a benchmark situation we can thus quantify

the univariate nature of the outliers by defining a new concept, called the Univariate-index, or *U-index*, as

$$\text{U-index} = \max_{i \in \{1 \ldots p\}} (|\text{corr}(x_i, y)|), \tag{6.13}$$

where corr is the correlation function, p is the number of features of the data set (without the outlier label y). In words, equation (6.13) says to take the maximum of the absolute value of the correlation coefficient of any feature with the binary label indicating the outliers. The index is added to Table 6.1 and will prove useful in Section 6.4.

6.3 Experimental Setup

6.3.1 General

To test the effectiveness of the algorithms mentioned in the previous section, the algorithms are applied to the benchmark data sets of Goldstein and Uchida [43]. The complete code of the experiments can be at [86]. In the experiments we used implementations of the algorithms as can be found in the R-packages: 'autoencoder' (version 1.0) [34], 'IsolationForest' (version 0.0-26) [70], and 'deepnet' (version 0.2) [100]. The latter package is used for the RBM and is adjusted slightly in order to implement a dynamic stopping criterion and the automatic adjustment of the learning rate, see Section 6.3.2.

The set of hyper-parameters that are tested for each algorithm will be explained in Section 6.3.2. For each set of hyper-parameters we will run ten experiments, each time with a different random seed. This will reduce the noise introduced by the random component that is present in all three algorithms. Reported results are averages over all runs. For instance, for an RBM we will test 16 different hyper-parameter settings (4 different number of hidden nodes times 4 different settings for the number of bins). A reported auc in Table 6.2 and Table 6.3 is thus an average over $4 \cdot 4 \cdot 10 = 160$ experiments.

Results are measured using the 'area under the curve' statistic, in line with the approach of Goldstein and Uchida [43].

6.3.2 Setting hyper-parameters

6.3.2.1 Sparse auto-encoder

In the experiments the most basic architecture of a sparse auto-encoder is tested, i.e., with one hidden layer. Since this simple architecture already achieved good res-

ults, see Section 6.4, we have not experimented with more complex variants of auto-encoders.

The main architectural hyper-parameter is the number of hidden nodes h. We choose $h \in \{5, 10, p/2, p\}$, where p is the number of features in the data set, excluding the label indicating the outliers. If $p/2$ is not an integer, we rounded downwards. An exception is made for the 'speech' data set that contains 400 features, requiring an exceptional long run time. For 'speech' we take $h \in \{5, 10, 25, 50\}$. The numbers 5 and 10 hidden nodes have been chosen since some initial experiments indicated that a reasonably low reconstruction error could be obtained with these numbers. The numbers $p/2$ and p are added to ensure that data sets with more features (more possible patterns) have a network with larger expressive power.

The learning rate λ is set to a value of 0.001 for all data sets, since this value ensured a smooth decreasing target function for all data sets. The sparsity parameter ρ is fixed to 0.1 as recommended by Ng in his lecture notes on auto-encoders [80].

6.3.2.2 Isolation forest

The isolation forest algorithm is robust concerning the values of its hyper-parameters. For all hyper-parameters we choose the values as recommended by Liu et al. [71].

6.3.2.3 Restricted Boltzmann Machine

The number of hidden nodes for the RBM is set equal to the values chosen for the sparse auto-encoder, i.e. $h \in \{5, 10, \lfloor p/2 \rfloor, p\}$ and $h \in \{5, 10, 25, 50\}$ for the 'speech' data set.

A classical RBM needs binary input. For this reason we pre-processed the data for RBM's by discretizing each feature into b bins, using equal width binning. Subsequently, a dummy variable is constructed for each bin, using thermometer encoding as mentioned in Section 6.2.3. In the experiments we set $b \in \{3, 5, 7, 10\}$.

The learning rate λ and the stopping criterion of the RBM is set dynamically. Initially $\lambda = 0.01$, subsequently λ is decreased by a factor 10 as soon as free energy of all observations between two epochs rises with more than 1 per cent. The algorithm is stopped as soon as the free energy of the data set changes less than 1 promille. The *free energy* serves as a proxy for the (log-) likelihood (6.10). The log-likelihood can be separated in two terms like,

$$\log \prod_{\mathbf{v} \in V} p(\mathbf{v}) = -\log Z(W, \mathbf{a}, \mathbf{b}) - \sum_{\mathbf{v} \in V} F(\mathbf{v}). \tag{6.14}$$

The last term is the free energy of the data set.

6.3.3 Data Sets

Table 6.1 provides a summary of the data sets of the benchmark of Goldstein and Uchida. [43]. All features in the data sets are numeric. Most data sets are originally posted for classification tasks. To make the data sets suitable for anomaly detection, typically observations from one specific class are labeled anomalies, while all other observations are considered normal cases.

As a data pre-processing step, a min-max scaling is applied to all features x of all data sets, resulting in a range of $[0, 1]$ for each feature,

$$x_{sc} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}}. \tag{6.15}$$

Below we will give a short description of each data set.

	data set name	number of rows	number of columns	outliers	percentage outliers	U-index
1	breast cancer	367	30	10	2.72	0.570
2	pen global	809	16	90	11.1	0.600
3	letter	1.600	32	100	6.25	0.193
4	speech	3.686	400	61	1.65	0.079
5	satellite	5.100	36	75	1.49	0.308
6	pen local	6.724	16	10	0.15	0.047
7	annthyroid	6.916	21	250	3.61	0.419
8	shuttle	46.464	9	878	1.89	0.675
9	aloi	50.000	27	1.508	3.02	0.029
10	kdd 1999	620.098	29	1.052	0.17	0.678

Table 6.1: Summary of the data sets used to compare the various anomaly detection algorithms. See equation (6.13) for the definition of the U-index.

6.3.3.1 Breast Cancer

This data set is derived from the Wisconsin Breast Cancer data set that contains medical data of 569 patients. The data consists of features derived from digitized images of breast mass, obtained via a Fine Needle Aspirate. In the original data set there are 357 patients with benign breast cancer and 212 patients with malignant breast cancer. In the data set prepared for anomaly detection, all benign patients are kept, while the first 10 patients with malignant breast cancer are labeled as anomalies.

6.3.3.2 Pen Global

The observations in this data set are feature vectors derived from images of the digit '8', handwritten several times by 44 different writers. The feature vectors have a length of 16 and contain eight (x, y) pairs. These pairs are positions that are recorded after fixed intervals when the digit is written. The anomalies are 10 observations from the digits '0', '1', '2', '3', '4', '5', '6', '7', '9' each, leading to 90 anomalies.

6.3.3.3 Letter

This data set consists of features extracted from 3 letters from the English alphabet. The outliers consist of the same features but extracted from the other letters of the alphabet. To make the anomaly detection task more challenging, the contributors added randomly some features to each observation, coming from all letters of the English alphabet.

6.3.3.4 Speech

This data set comes from the domain of speech recognition. Each observation is a so-called 'i-vector representation' of a speech segment. The normal cases come from persons with an American accent, while outliers consist of persons with other accents.

6.3.3.5 Satellite

This data set consists of features extracted from satellite images. These images are used to determine the soil type. In this data set the soil types: 'red soil', 'gray soil', 'damp gray soil' and 'very damp gray soil' are normal instances. Anomalies were sampled from the classes: 'cotton crop' and 'soil with vegetation stubble'.

6.3.3.6 Pen Local

This data set has the same underlying data set as 'Pen Global'. However, instead of focusing on the digit '8', all digits are kept with the exception of the digit '4'. From the latter digit only the first ten observations are included and these form the anomalies.

6.3.3.7 Annthyroid

This data set is derived from the annthyriod data set, also known as the Thyroid disease data set, and comes from the medical domain. It contains features from patients; normal cases represent healthy patients, the outliers are sampled from the patients

that suffer from hypothyroid cancer. The first fifteen features are binary features, the next six features are continuous.

6.3.3.8 Shuttle

This data set is used in the Statlog project and contains features that are connected to the normal and abnormal functioning of radiators in a NASA space shuttle. The original data set is designed for supervised anomaly detection. The version used in this chapter is adjusted by Goldstein and Uchida [43] mainly by reducing the number of outliers.

6.3.3.9 ALOI

The aloi data set originates from a data set provided by the Amsterdam Library of Object Images (ALOI). This library contains images of objects. This particular data set contains a feature vector of length 27 for each image, derived by apply a HSB color histogram. Such a histogram gives the distribution of colors in an image. Each object is photographed many times under different angles and lighting conditions. The 1.508 observations labeled as anomalies correspond to a few objects selected as anomalies.

6.3.3.10 KDD Challenge 1999

This data set comes from a challenge presented at the Knowledge Discovery and Data Mining conference of 1999. The data set contains artificially created observations that represent HTTP traffic in a computer network. The data set is enriched with observations representing observations typically seen in attacks. The current data set has undergone some data preparations to make it more suitable for testing various anomaly detection algorithms, see [43] for details.

6.4 Results

Table 6.2 summarizes the findings of the experiments. The isolation forest algorithm realizes the highest area under the curve on the ‘shuttle’ data. The RBM reaches first place for ‘breast cancer’ and ‘kdd 1999’, although the first position is shared with HBOS (Histogram-Based Outlier Score) for the latter data set. The auc-values for all algorithms in the benchmark can be found in Table 6.3.

Figure 6.4 displays the overall performance of the algorithms on all data sets. The k-NN and kth-NN algorithms perform the best in general. Only on the ‘annthyroid’

	data set	auto-encoder	isolation forest	RBM	mean bench-mark	best bench-mark	best alg. bench-mark
1	breast cancer	0.9091 ± 0.0040	0.9810 ± 0.0014	**0.9858** ±0.00005	0.9067	0.9827 ± 0.0016	HBOS
2	pen global	0.9420 ± 0.0008	0.9304 ±0.0016	0.8282 ±0.0014	0.7836	0.9872 ± 0.0055	k-NN
3	letter	0.7667 ± 0.0042	0.6337 ±0.0052	0.5794 ±0.0026	0.7850	0.9068 ± 0.0078	LoOP
4	speech	0.4716 ± 0.0002	0.4699 ±0.0052	0.4715 ± 0.0002	0.4936	0.5347 ± 0.0343	LoOP
5	satellite	0.9057 ± 0.0017	0.9479 ±0.0018	0.9060 ±0.0018	0.8734	0.9701 ± 0.0007	k-NN
6	pen local	0.8346 ± 0.0039	0.7828 ±0.0064	0.8220 ±0.0036	0.9129	0.9816 ± 0.0024	LOF
7	ann-thyroid	0.5657 ± 0.0010	0.6456 ±0.0058	0.5089 ±0.0026	0.6312	0.9150 ± 0.0123	HBOS
8	shuttle	0.9881 ± 0.0000	**0.9973** ±0.0002	0.9832 ± 0.0004	0.7684	0.9925 ± 0.0039	rPCA
9	aloi	0.5415 ± 0.0004	0.5408 ±0.0003	0.5311 ±0.0006	0.6229	0.7899 ± 0.0093	LoOP
10	kdd 1999	0.9718 ± 0.0001	0.9656 ±0.0017	**0.9990** ±0.00004	0.7926	0.9990 ± 0.0007	HBOS

Table 6.2: Mean Area Under the Curve (AUC) and standard deviation when applying the anomaly detection algorithms with various settings on the benchmark data sets.

data, these algorithms perform below average. The main characteristic of the 'annthyroid' data are its binary features. These binary features have almost no relation with the outlierness of an observation, but do have a large influence on the distance-based k-NN and kth-NN algorithms. Hence we may say that for distance based methods, scaling is important and may give problems when combining binary (or categorical) features with continuous ones.

The Cluster-Based Local Outlier Factor (CBLOF) is clearly the worst algorithm, although variants of this algorithm (uCBLOF and LDCOF) clearly improve performance considerably. The uCBLOF algorithm even reaches the third place in the ranking of algorithms in Figure 6.4. However it performs always worse than the simpler k-NN and kth-NN algorithms, except for data sets with a large U-index (shuttle, kdd 1999). Here the underlying clustering approach may have its advantages.

The Histogram-Based Outlier Score (HBOS) algorithm and the Local Outlier Probability (LoOP) algorithm deserve attention as well, since they are the top performers

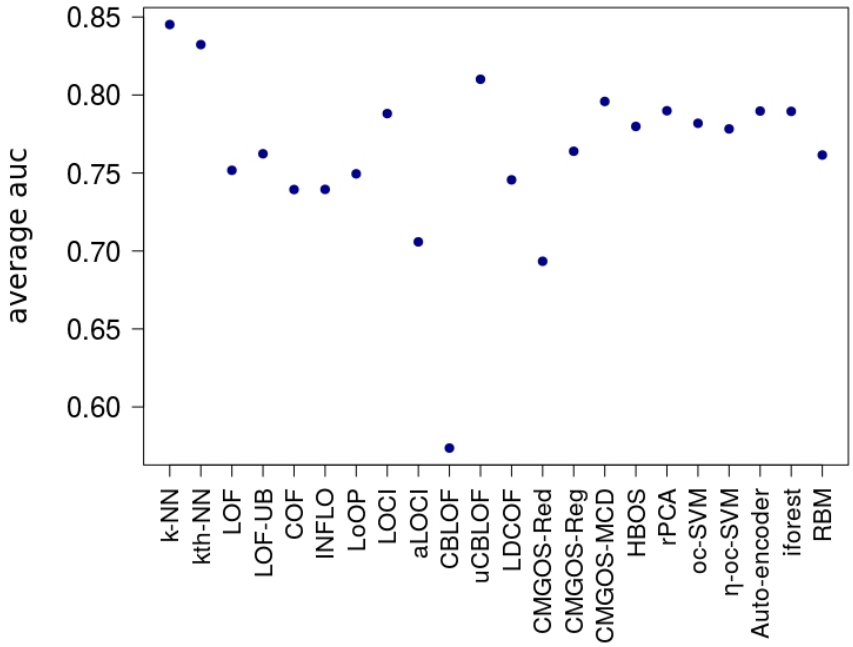


Figure 6.4: Average Area Under the Curve of all algorithms of the benchmark and the three newly added algorithms. The average is taken over all data sets in the benchmark.

for 6 of the 10 data sets among the original 19 algorithms of the benchmark. The simple HBOS algorithm is clearly strong on data sets with a large U-index (an indication for univariate outliers) like breast cancer, shuttle and kdd 1999, while weak on data sets with a small U-index (aloi, letter). It performs also well on annthyroid, where it does not get distracted by the binary features that have almost no correlation with the outliers. In contrast to HBOS, the LoOP algorithm performs well for data sets with a small U-index (letter, pen local, speech, aloi), but badly on data sets with a large U-index (shuttle, kdd 1999).

If we now turn our attention to the newly added algorithms, then we see that all three algorithms are good in finding outliers in data sets with a large U-index (shuttle, kdd 1999, breast cancer, pen global and satellite), often surpassing the currently best algorithm. However the new algorithms perform badly on data sets with a small U-index (letter, speech, aloi). The isolation forest algorithm can best handle the binary features that are present in annthyroid. Because of this property and the fact that isolation forest has the shortest run times and requires almost no tuning of hyperparameters, this algorithm could be labeled as the preferred choice between the three

new algorithms.

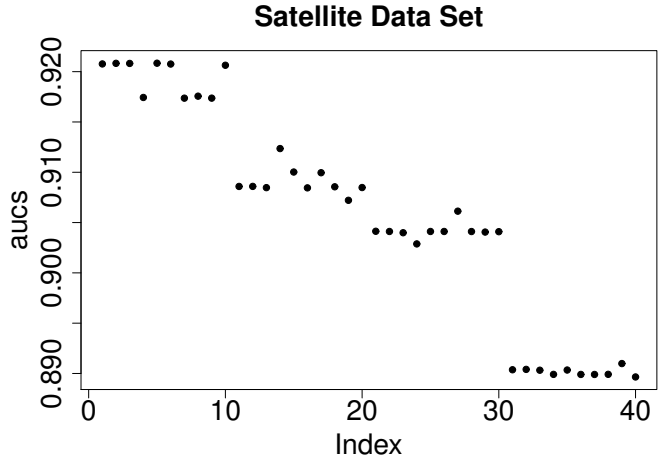


Figure 6.5: Value of the Area Under the Curve for the auto-encoder on the 'Satellite' data.

The time complexity for training one epoch and scoring the data is linear in the number of observations n, the number of features p and the number of hidden nodes h ($\mathcal{O}(nph)$) for standard sparse auto-encoders and RBMs. The time to train and score an isolation forest is linear in the number of observations and independent of the number of features ($\mathcal{O}(n)$). In practice, the run time of the algorithms is to a large extent dependent on the number of epochs needed before reaching convergence during training. With respect to this, isolation forest is the fastest algorithm. From the remaining two, RBMs reached convergence sooner than auto-encoders in our experiments. However this may be caused to a large extent by the use of the BGFS algorithm to optimize the target function for auto-encoders instead of the generally faster stochastic gradient descent method.

After running the experiments for auto-encoders, we have plotted the number of hidden nodes against the auc in order to get more insight in the number of hidden nodes needed in the context of anomaly detection. One of these plots is displayed in Figure 6.5. Observations with index 1 to 10 in this plot are coming from auto-encoders with 5 hidden nodes, index 11 to 20 with 10 hidden nodes, index 21 to 30 with 18 hidden nodes, and finally, index 31 to 40 with 36 hidden nodes. All other hyper-parameters are fixed (except for the random seed that changes for each run). It is clear from the graph that auto-encoders with a small number of hidden nodes are better able to find the outliers (larger auc value). Also for the plots of the other data sets, it is clear that for most data sets 5 or 10 hidden nodes are sufficient.

Algorithm Alg.	breast- cancer	pen- global	pen- local	letter	speech	satel- lite	thy- roid	shuttle	aloi	kdd 1999
k-NN	0.9791	**0.9872**	0.9837	0.8719	0.4966	**0.9701**	0.5956	0.9424	0.6502	0.9747
kth-NN	0.9807	0.9778	0.9757	0.8268	0.4784	0.9681	0.5748	0.9434	0.6177	0.9796
LOF	0.9816	0.8495	**0.9877**	0.8673	0.5038	0.8147	0.647	0.5127	0.7563	0.5964
LOF-UB	0.9805	0.8541	0.9876	0.9019	0.5233	0.8425	0.6663	0.5182	0.7713	0.5774
COF	0.9518	0.8695	0.9513	0.8336	0.5218	0.7491	0.6505	0.5257	0.7857	0.5548
INFLO	0.9642	0.7887	0.9817	0.8632	0.5017	0.8272	0.6542	0.493	0.7684	0.5524
LoOP	0.9725	0.7684	0.9851	**0.9068**	**0.5347**	0.7681	0.6893	0.5049	**0.7899**	0.5749
LOCI	0.9787	0.8877		0.788	0.4979					
aLOCI	0.8105	0.6889	0.8011	0.6208	0.4992	0.8324	0.6174	0.9474	0.5855	0.6552
CBLOF	0.2983	0.319	0.6995	0.6792	0.5021	0.5539	0.5825	0.9037	0.5393	0.6589
uCBLOF	0.9496	0.8721	0.9555	0.8192	0.4692	0.9627	0.5469	0.9716	0.5575	0.9964
LDCOF	0.7645	0.5948	0.9593	0.8107	0.4366	0.9522	0.5703	0.8076	0.5726	0.9873
Red	0.914	0.5693	0.9727	0.7711	0.5077	0.9054	0.4395	0.5425	0.5852	0.7265
Reg	0.8992	0.6994	0.9449	0.8902	0.5081	0.9056	0.6587	0.5679	0.5855	0.9797
MCD	0.9196	0.6265	0.9038	0.7848		0.912	0.8014	0.6903	0.5547	0.9696
HBOS	0.9827	0.7477	0.6798	0.6216	0.4708	0.9135	**0.915**	0.9925	0.4757	**0.999**
rPCA	0.9664	0.9375	0.7841	0.8095	0.5024	0.9461	0.6574	0.9963	0.5621	0.7371
oc-SVM	0.9721	0.9512	0.9543	0.5195	0.465	0.9549	0.5316	0.9862	0.5319	0.9518
η-oc-SVM	0.9581	0.8993	0.9236	0.7298	0.4649	0.943	0.5625	0.9848	0.5221	0.7945
Auto-enc.	0.9091	0.942	0.8346	0.7667	0.4716	0.9057	0.5657	0.9881	0.5415	0.9718
iforest	0.981	0.9304	0.7828	0.6337	0.4699	0.9479	0.6456	**0.9973**	0.5408	0.9656
RBM	**0.9858**	0.8282	0.822	0.5794	0.4715	0.906	0.5089	0.9832	0.5311	**0.999**

Table 6.3: Mean Area Under the Curve for all data sets and algorithms in the benchmark.*Red* stands for CMGOS-Red, *Reg* stands for CMGOS-Reg, *MCD* stands for CMGOS-MCD

6.5 Conclusions and Discussion

Table 6.3 shows the performance of all algorithms on the benchmark of Goldstein and Uchide [43], including the three newly added algorithms. A main observation is that the three algorithms are able to meet or beat the current best algorithm in the benchmark several times: isolation forest is superior on the 'shuttle' data set, while the RBM outperforms all current algorithms on 'breast cancer' and matches the best performance on the 'kdd 1999' data set, see Table 6.2.

Isolation Forest seems to be the preferred choice of the three algorithms that are newly added to the benchmark; the area under the curve on the benchmark is rather similar to the other two algorithms, but it has shorter run times and its hyper-parameters are easy to set. Moreover it handles the binary features in the 'annthyroid' data set well.

Another observation is that there are (at least) two types of data sets with outliers: in the first type of data sets there is at least one feature that has a correlation with the label indicating an outlier. Anomaly detection algorithms that perform well on these data sets are: HBOS, rPCA, oc-SVM, η-oc-SVM, auto-encoder, isolation forest, RBM, and uCBLOF. We see that all three new algorithms fall in this category. The second type of data sets lack such a univariate feature. On these data sets LOF-like (Local Outlier Factor) algorithms perform well: LOF, LOF-UB, COF, INFLO, and LoOP. The U-index, as introduced by equation (6.13), helps to classify the data sets in the benchmark in these two groups, see Table 6.1.

Further, it is noteworthy that simplicity seems to go hand in hand with power at several places. This is evident in Figure 6.4, where it becomes clear that the simple algorithms of k-NN and kth-NN perform the best when considering the average performance on all data sets. Also, Table 6.2 shows that the relatively simple HBOS algorithm belongs to the top performers for data sets with a large U-index. Finally, we note that simplicity in the number of nodes (i.e. a small number) for auto-encoders and RBMs does not decrease performance, see for instance Figure 6.5.

When reflecting on the results, then some reservations are in order. First, the benchmark contains a limited number of data sets (ten), and a disproportional large number of these data sets are features extracted from images ('breast cancer', 'pen global', 'pen local', 'letter', 'satellite', 'aloi'). Moreover all data sets are tabular data and contain no categorical features. Second, in this chapter only the most common implementations of the three newly added algorithms are tested. Each algorithm knows extensions that are worth further investigation in the future: the isolation forest algorithm has recently been extended by allowing splits in the feature space that are not parallel to coordinate axes, see the paper of Hariri et al. [48]. Auto-encoders can be extended by allowing more hidden layers, while also interesting variants exist that

are mainly developed for large data sets (variational auto-encoders, GAN-networks [103]). RBMs can be stacked, leading to deep belief networks. Also a combination of auto-encoders and RBMs exists, see the paper of Hinton and Salakhutdinov [52]. Here RBMs initialize the weights in deep auto-encoders, making it feasible to train these networks with gradient descent.

For further research we also want to mention the 'speech' data set. None of the current algorithms performs well on this data set due to the combination of many features (400) and a low U-index.

7 Various Topics

Besides the four main topics treated in the previous chapters, we have touched four others as well. We have included these contributions in a separate chapter. We hope that by seeing these various topics, the reader gets an impression of the broadness of the application field of data science to the administration of taxes.

We start with an application of analytics to Human Resources. This application area is also known as *HR Analytics*. In the section, a model will be constructed to explain differences in time allocation of employees. Then we will look at the connection between reinforcement learning and tax collection. Reinforcement learning forms a large domain of machine learning, next to *supervised learning* and *unsupervised learning*. Reinforcement learning has attracted a lot of attention lately, when the combination of deep learning and reinforcement learning proved to be able to beat the world champion of the board game Go. In the section we look at its potential to improve choices in the the collection of tax debts. In the third section, the focus is on a way to explain risk models to non-experts. Explaining complex algorithms from machine learning is often essential to introduce these techniques in a tax administration. Finally, we look at some ideas from Fuzzy Sets, and apply these on tax data, in line with the approach of Meza et al. [76]. The application on real data gives evidence of the predictive power of this approach.

Some parts of the chapter have been published in the following articles:

- C. Boon, F. D. Belschak, D. N. Den Hartog, and M. Pijnenburg. Perceived human resource management practices. *Journal of Personnel Psychology*, 2014

- M. Pijnenburg and K. Kuijpers. Explaining risk models to the business. *Tax Tribune*, 35:57–62, 2016

Besides, the idea about applying reinforcement learning has been worked out in the following Master Thesis, supervised by the author,

- M. Post. Tax data and reinforcement learning. Master's thesis, Leiden University, 2 2019. Under supervision of Mark Pijnenburg, Wojtek Kowalczyk, and Kaifeng Yang.

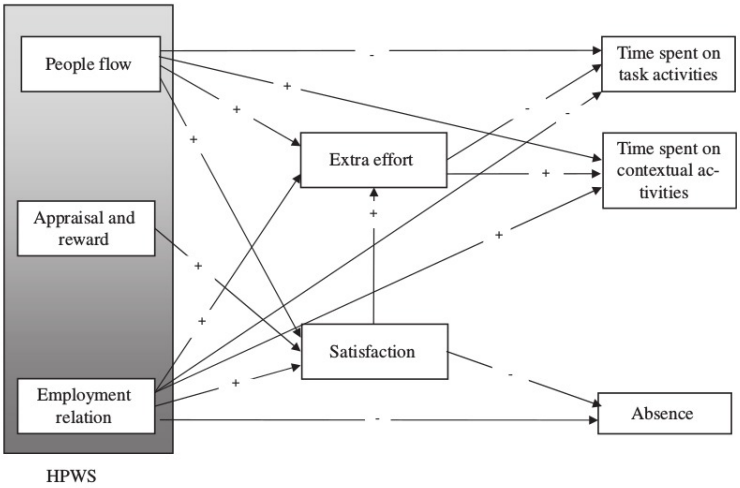


Figure 7.1: Model for the relation between (perceived) HR practices and time allocation

7.1 Perceived Human Resources Practices and Time Allocation

In a company, a difference can be made between 'core processes' and 'support processes'. Core processes deliver the products or services that are sold by the company. Support processes assist these core processes. Examples of support processes are financial processes and human resource (HR) processes.

Analytics can be applied to support processes as well. Applications of analytics to HR are frequently called *HR Analytics*. As an illustration of HR Analytics at the Netherlands Tax and Customs Administration, we present the use of confirmatory factor analysis to verify a hypothesized model. This model establishes a relationship between the time spent by employees and the (perceived) quality of HR practises. A full description of this research can be found in [18].

In short, based on available knowledge in the HR literature, a relation between perceived HR practices and time allocation of employees is hypothesized, as depicted in Figure 7.1. At the right side of the figure, three 'HR bundles' are shown, *people flow*, *appraisal and reward*, and *employment relation*. These three bundles, that relate to different HR aspects, can be seen as the components of a High-Performance Work System (HPWS). The three HR bundles are expected to influence the time allocation of employees. As can be seen from the Figure (left side), the time allocation is split into three components: time spend on task activities, time spend on contextual

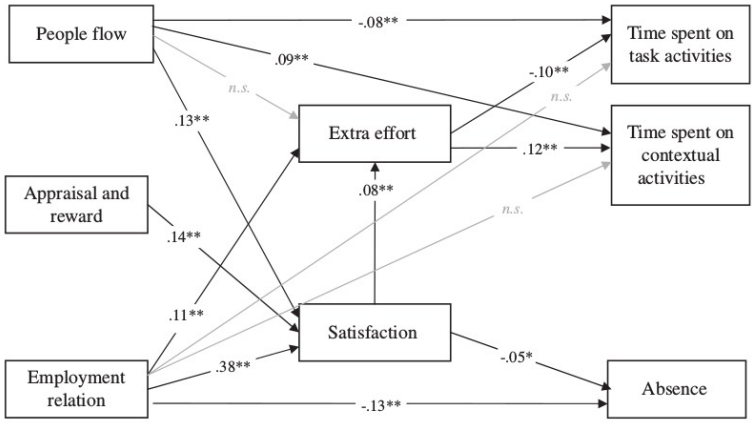


Figure 7.2: Result after applying confirmatory factor analysis

activities, and absence (mainly due to illness). Task activities are tasks mentioned in the job description and related to an internal or external client. Contextual activities are other activities like networking and training, that are necessary to guarantee future production. The influence of the HR bundles can be directly, or indirectly by the concepts of *extra effort* and *job satisfaction*.

The hypothesized relation can be tested by applying confirmatory factor analysis, a standard technique from statistics. Figure 7.2 shows the significance of the hypothesised relations after testing. Clearly, three hypothesised relations are not significant, while the others are. These insights may be used to optimize the HR practices at the NTCA or other organizations, in order to diminish unwanted absence or the increase employee satisfaction or time spent on task activities or contextual activities.

7.2 Reinforcement learning applied to tax debt collection

Reinforcement Learning is sometimes described as the third central pillar of learning from data, next to supervised learning and unsupervised learning [78]. *Supervised learning* is characterized by the presence of a sample of data for which we have additional information. Usually, the task of supervised learning is to learn the additional information for cases outside the sample. With *unsupervised learning* such additional information is lacking, and the task is usually to find structure in the data set, like the probability distribution of observations, or the location of clusters or anomalous observations.

Reinforcement learning is different from the previous two concepts, since there is no classical tabular data set with observations and features. Instead, in reinforcement learning, there is a player that is in a certain state (of an environment) who can take a limited number of actions. By taking an action, the state changes and there may occasionally follow a reward or punishment. By taking actions and observing the change of the states the player tries to understand the environment and to find a sequence of actions that will optimize his rewards.

Although the description of reinforcement learning may sound overwhelmingly at first, it is really close to human learning. Consider an example: when a child (the player) learns to walk, it is at a certain position (state) in a room (the environment). When it does some muscle movements (actions) it may change position and get some reward; a positive reward if it experiences that it has walked some distance, and a negative if it bumps against an object or falls. By reflecting on the muscle movements and the change in the state that followed these movements, the child slowly starts to understand how to interact with the environment it is in. Moreover, eventually it will learn what muscle movement will give good rewards: it has learned to walk! Another example is chess play; when a player has done an action (moved a piece), the state of the environment (the positions on the chessboard) has changed and at the end a positive (win) or negative (loss) reward occurs. By observing the effects of his actions, the player may learn how to play chess. A comprehensive introduction to reinforcement learning is the book of Sutton [105].

Reinforcement learning obtained some spectacular results recently. Supported by deep learning technologies, a reinforcement learning algorithm called AlphaGo was able to defeat the world champion 'Go' in March 2016. AlphaGo's successor AlphaZero is able to learn the games of chess, shogi, and go from scratch within a day and reach a superhuman level. In December 2017 AlphaZero defeated the best computer programs for these three games.

The debt collection process of a tax administration resembles the basic set-up of reinforcement learning: open tax debts may follow a sequence of actions taken by the tax administration (the player) starting with a reminder and usually ending with payment. Each action of the tax administration may lead to another action of the debtor, for instance a phone call of the tax administration may lead to a formal request for deferment from the debtor. As a result, a sequence of actions occurs, where each time the state of the debt may change. Rewards occur when the debt is (partially) paid, while a loss occurs (for the tax administration) when the debt is written off. Hence the idea of applying reinforcement learning to learn an optimal strategy for the tax administration to collect unsettled debts.

Actions that the tax administration can take include calling the taxpayer, offering a payment plan, initiating a pay claim, and allow a delay of payment. Currently, at

the NTCA, the tax administration usually starts with mild measures that are getting harsher if it becomes clear that the taxpayer is unwilling to pay. If the taxpayer is unable to pay, the tax administration tries to take a more supportive position.

More on applying reinforcement learning on the collection process of a tax administration can be found in the master thesis of Martijn Post [95], written under the author's supervision. In summary, Post implemented several variants of a simple reinforcement algorithm to an anonymous data set of the NTCA (Dutch tax administration). The algorithm produces a value function for the current policy that gives sensible values for possible next actions. To improve on these initial results, more data of the taxpayers has to be added, like their ability to pay. Moreover, the algorithm must be able to explore the environment by taking some unconventional actions from time to time. Both these extensions were thought to be undesirable with a view on (the privacy of) the taxpayers and out of proportion. Especially since the current collection process can probably be improved with less far-reaching analyses. For these reasons we have not continued this line of research.

7.3 Explaining risk models to non-experts

Risk models are becoming more popular in tax administrations for audit selection. Although risk models are not new to tax administrations, only recently they are becoming mainstream and are embedded in central business processes. The fact that an, often complex, computer algorithm is at the heart of a risk model makes it harder to explain to auditors and managers, who use or are responsible for the model. A good explanation is an integral part of getting these new, often efficient, techniques accepted.

Tax administrations are most familiar with risk rules, like "if this year's amount differs more than 20% from previous year's amount, then select for audit". Domain experts construct these rules. In contrast, risk models are created by applying a computer algorithm to historical data. Note that in practice, often, a combination of risk rules and risk models is applied.

An analogy can help explain a risk model. Below we present an analogy that has been tested by the Netherlands Tax and Customs Administration. The analogy explains as well some essential concepts such as 'Analytical Base Table' and 'target'.

To begin with, auditors and managers are asked to imagine that they are teachers in a class of students. As a teacher they are interested in knowing what students failed to complete their homework. Of course, they can check every student, but this will consume most of the lesson, leaving no time to teach. Just like auditors who only have limited capacity to control a large number of companies.

The analogy is continued by assuming that the teacher decides to choose three students for each lesson to be checked for their homework. The teacher wants to check the three students with the largest chance of not having completed their homework. To select these students, the teacher uses different *features* of his students, see Figure 7.3. These features can be any characteristic of a student, such as sitting in the back of the classroom, gender, results of previous homework checks, and perceived kindness. In the context of taxation, features are usually characteristics of taxpayers or tax returns and we want to check the correctness of tax returns.

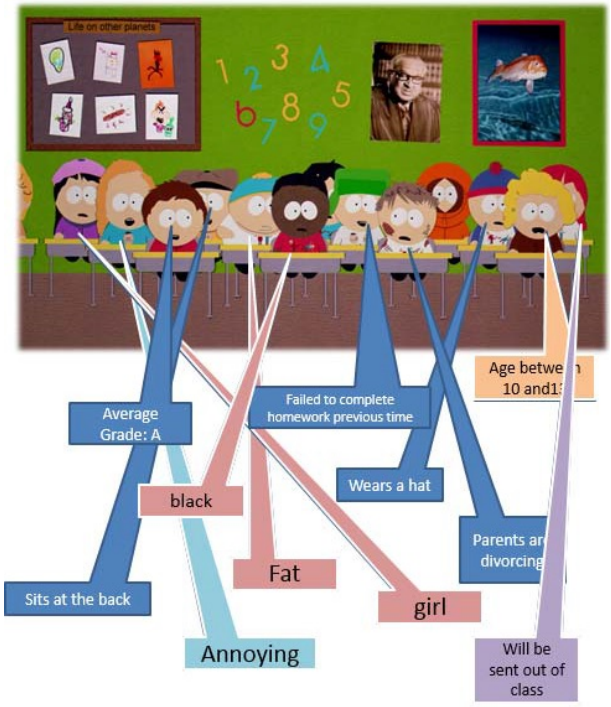
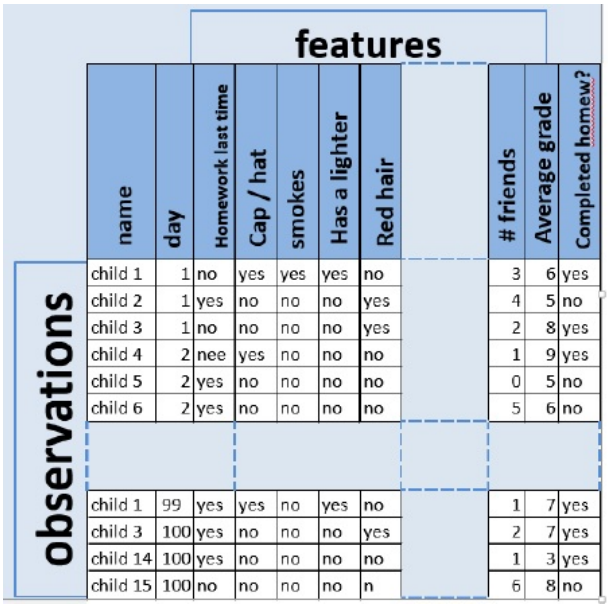


Figure 7.3: Explaining a risk model by analogy of a classroom.

Then it is explained that not all features are suitable for the selection of students. Some features might be in conflict with the way we want to treat students; it is usually not desired to discriminate between students (or taxpayers) based on features like race or gender. Moreover, some features cannot be determined objectively; a feature like 'annoying' is not objective since a student might be annoying for one teacher, but not for another. Such a feature can thus only be used for a risk model for that teacher, not for all teachers. A third class of features that are not suitable for selection are features whose value is unknown at the time we want to apply the model. This may seem obvious, but a risk model is usually build on historical data. We are thus

in the future with respect to the date the data was created. Therefore, we may have information that was unknown at the time the data was created (e.g. the taxpayer has raised a dispute). In our student example, the fact that a student is expelled from the class is an example of such a feature.

After selecting suitable features, the teacher can start building a data set, called *Analytical Base Table* (ABT), see Figure 7.4. This means that for a while, the teacher



name	day	Homework last time	Cap / hat	smokes	Has a lighter	Red hair	# friends	Average grade	Completed homew?
child 1	1	no	yes	yes	yes	no	3	6	yes
child 2	1	yes	no	no	no	yes	4	5	no
child 3	1	no	no	no	no	yes	2	8	yes
child 4	2	nee	yes	no	no	no	1	9	yes
child 5	2	yes	no	no	no	no	0	5	no
child 6	2	yes	no	no	no	no	5	6	no
child 1	99	yes	yes	no	yes	no	1	7	yes
child 3	100	yes	no	no	no	yes	2	7	yes
child 14	100	yes	no	no	no	no	1	3	yes
child 15	100	no	no	no	no	n	6	8	no

Figure 7.4: Building an Analytical Base Table for the student analogy of a risk model.

registers features of each student who has been checked. Each student receives a row in the Analytical Base Table and each column contains a feature of the students. The last column indicates whether or not the student has completed the homework. This column, often called the *target*, contains the information that we want to learn to predict. The Analytical Base Table is the starting point for creating a risk model.

The next step is to construct a model. Although in a previous step we have selected a set of suitable features, not all features will have predictive power. So the model building process is started with *feature selection*, i.e. selecting the most promising features. In the class example, a teacher may start looking at the values of the individual features in the Analytical Base Table and compare them with the target. The teacher can note that the feature of low average grades often goes together with the fact that the homework has not been completed. There might be no such relation between the target and a feature like 'wearing a hat'. This way, the teacher comes to a shortlist of features that are thought to be promising for selecting risky students.

Now, selected features have to be combined to come to a final risk score for each student. Although many ways exist to do this, in the analogy we take the approach of assigning individual risk points to each feature and then compute the final score by adding up these risk points. In the student's example we can, for instance, assign two risk points to the student if the homework was not completed the last time and one risk point when the average grade is low. The total risk score is then 3. The determination of the right risk points for each feature is an essential part for obtaining a good model. We then demonstrate a brute force way that tries a large collection of different risk points and finally selects the one that gives the best result on the Analytical Base Table.

Finally the scoring of new students with the help of this risk model is demonstrated, visually. Here the model is represented as a machine that gets as input the features of a new student and returns a risk score This then enables to rank the students based on risk. The three students with the highest scores are eventually checked by the teacher.

7.4 Recommendations based on fuzzy sets

Recently, Meza et al. [76] propose to use fuzzy sets and recommendation systems to improve taxation. Their idea is to represent some key concepts for taxation, in particular the compliance of a taxpayer and the riskiness of tax debt, as fuzzy sets. Subsequently, a tax administration may adapt its approach based on the membership degrees to these fuzzy sets. The approach has been tested on small scale by the authors for the municipal taxation of the city of Quite (Ecuador).

Although the ideas about the applicability of fuzzy sets for tax administrations are still developing, we have tested some aspects of it at the Netherlands Tax and Customs Administration, in cooperation with the authors of [76]. Two aspects have been investigated: first the applicability of the fuzzy sets approach. This resulted in Figure 7.9. Second, the predictive power of the approach, resulting in Table 7.1. Both investigations are explained below.

To test the framework, we selected 100,000 Dutch tax debts that were open on 1 July 2015 and were fully completed by 1 August 2018. We have ensured that all selected debts were before the phase that a judicial notice had to be sent. We selected these 100,000 debts in such a way that 50,000 were paid before the payment deadline (or a payment arrangement was requested), and 50,000 at least 10 days after the deadline. In all other respects, the debts are selected at random.

For each debt, we looked at the citizen involved and constructed a Citizen Behavior Ranking. We looked at historic payment data of the citizen in the period from 1

July 2013 to 1 July 2015. For each debt of the citizen that was closed in that two-year period, we recorded whether the debt was paid before the payment deadline or not. Then we calculated the ratio of the number of debts that were paid in time to all debts. Figure 7.5 shows a histogram of the CBR for our population belonging to the 100,000 debts.

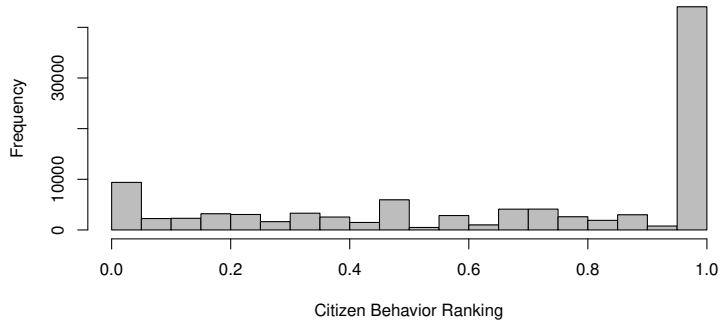


Figure 7.5: Citizen Behavior Ranking for the 100,000 citizens belonging to the 100,000 debts

Based on the Citizen Behavior Ranking, fuzzy sets for Citizen Behavior can be easily created. For instance by using Figure 7.6. Applying the membership functions of Figure 7.6, we find that a citizen with a Citizen Behavior Ranking of, say, 0.85 has a membership degree to the fuzzy set 'Excellent' of 0.4, a membership degree of 0.6 to the fuzzy set 'Good' and 0 for all other fuzzy sets.

Note that to calculate the Citizen Behavior Ranking we made use of a Central Data Repository that is daily refreshed using ETL processes. To respect the privacy of the taxpayers involved, the analysis has been performed using data where all identifying records have been removed.

For the 100,000 debts mentioned before, we calculated the days till the payment deadline. In some cases this number was negative, which means that the payment deadline had been passed already. Inspired on Figure 3 of [76], here reproduced for reference as Figure 7.7, we come to a membership degree for the fuzzy set 'Payment Deadline'. A membership degree of 1 is given to debts that were over the payment deadline and for debts that were before the deadline a linear relation was used, starting with a membership degree of 0 on the day the debt was communicated to the citizen.

Similarly, a membership degree for the fuzzy set 'risk of debt' is computed. We

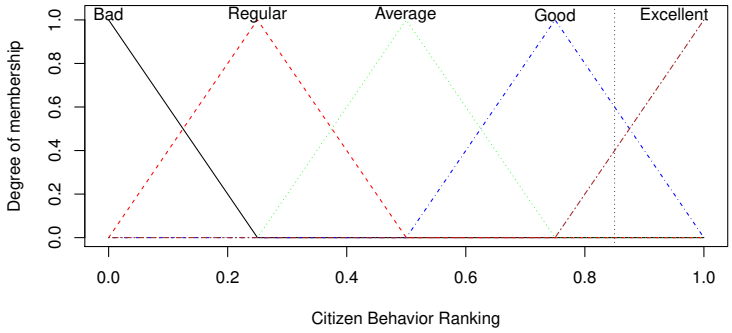


Figure 7.6: Fuzzy sets for Citizen Behavior for the Dutch Citizens belonging to the 100,000 selected debts.

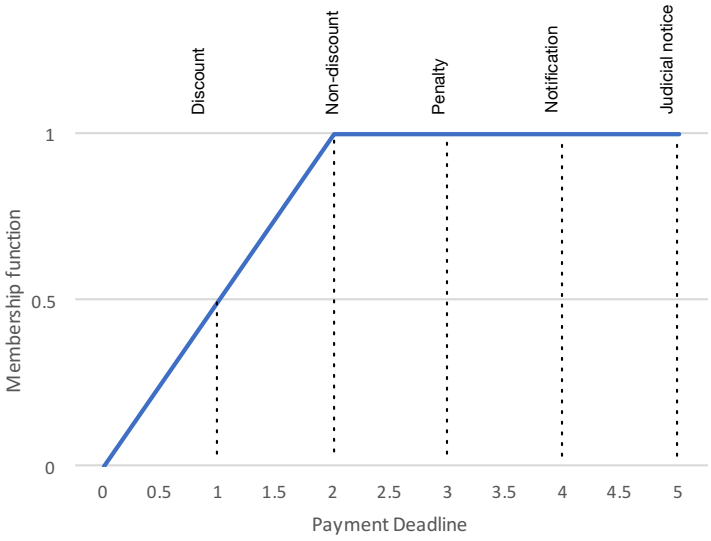


Figure 7.7: Reproduction of Figure 3 of [76], showing the relation between the payment deadline of a tax debt and the membership function of a fuzzy set related to late payment.

ordered the amounts of the 100,000 debts and took the rank number and divided that by 20,000 in order to get a risk relevance scale between 0 and 5. Subsequently, we applied the curve shown in Figure 4 of [76], here reproduced for reference as Figure 7.8, to come to a membership degree to the fuzzy set 'risk of debt'.

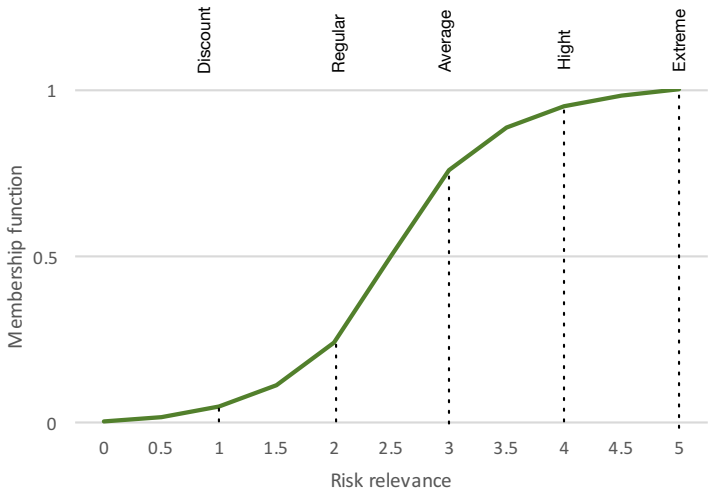


Figure 7.8: Reproduction of Figure 4 of [76], showing the relation between the risk relevance and the membership function of a fuzzy set related to extreme riskiness payment.

The Citizen Behavior Ranking, membership functions of 'payment deadline', and 'risk of debt' can be plotted in two two-dimensional Figures as shown in Figure 7.9. Note that we have plotted a random sample of size 200 of all 100,000 debts. The dashed line separates the risky debts from the less risky debts. It is up to the management of the tax authority to define this line. After the separation, different treatments (recommendations) for the risky and non-risky debts can be considered. For instance, it can be decided to do nothing for the non-risky debts, while the risky debts for payment deadline are sent a reminder letter, while the risky debts for risk relevance are receiving additional monitoring.

To assess the predictability of the separations in Figure 7.9, we focused on the plot at the top of Figure 7.9, i.e. on the risk of non-payment before the payment deadline. The 100,000 debts are separated a priori according to the separation line. Subsequently, it was checked a posteriori whether the debts were actually paid before the payment deadline. This gives the confusion matrix shownn as Table 7.1. The Chi square test for independence has been applied to the confusion matrix, resulting in a value of $9,104.6$ for the chi square statistic. Since the degrees of freedom is equal to

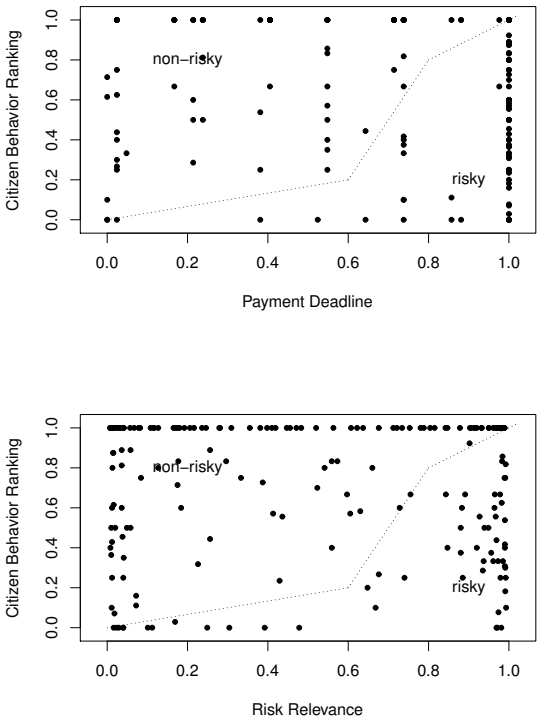


Figure 7.9: Citizen Behavior Ranking versus Payment Deadline and Risk Relevance for a random sample of 200 debts. The dotted lines separate risky debts from non-risky debts.

A Priori Risk	Paid	Not Paid
Not Risky	38,830	24,269
Risky	11,170	25,731
Total	50,000	50,000

Table 7.1: Confusion matrix. Riskiness versus Paid on time

1, this leads to a p-value of $2.2 \cdot 10^{-16}$, signifying a clear dependence between a priori risk predictions and actual payment problems.

8 Conclusion

In previous chapters, some selected topics in the field of data science and taxation have been treated in detail. These selected topics are limited in their scope, but can be considered as small bricks, that will allow science eventually to construct a solid building. In this chapter, we will look again at the conclusions of the various chapters, and try to place them in a wider context. To do so, we will look at each chapter individually and pay special attention to the research question underlying the chapter. After that, we will take a step back and reflect shortly on the future perspective of data science and taxation.

8.1 Analytics in Taxpayer Supervision

The research of this chapter started with the research question

> **Research Question** *What data science / analytical techniques can be used in* taxpayer supervision *and what contributions may be expected from these techniques?*

A list of the techniques that can be used can be found in Table 2.4. Concerning the expectations, the main contribution of the chapter is to bring realism in the accomplishments that can be expected from applying analytics to taxpayer supervision. Analytics will not replace the paradigm of Compliance Risk Management in taxpayer supervision, but has potential to support many current supervision activities within that framework. So, neither the claim that analytics is just a hype and will hardly contribute to taxpayer supervision, nor the claim that analytics will fundamentally change taxpayer supervision is supported.

The conclusion that analytics complements current activities, instead of replacing them, seems to have plausibility for other domains as well. Think for example about internal processes of tax administrations or other supervision tasks of governments.

At the same time, it is essential not to generalize this conclusion outside the realm of analytics. If we think of data science in general (recall Figure 1.7 for the difference between analytics and data science in this thesis), this also contains the field of Artificial Intelligence (AI) that is currently making fast progress towards reproducing certain aspects of human intelligence. Technologies developed in AI have potential for changing essential aspects of our society. And if society will change, taxation will be adjusted to it as well.

Figure 8.1 reflects our believes about the direct and indirect influence of developments in data science on taxation. We believe that the direct application of data science by tax administrations to improve its processes will most likely lead to more effectiveness (and some gain in efficiency), without changing fundamental paradigms (blue line in Figure 8.1). This is in line with the conclusions of chapter 2. In contrast, applications of data science in the wider society may change some fundamental aspects of society, and thereby indirectly changing fundamental aspects of taxation (brown dotted line in Figure 8.1).

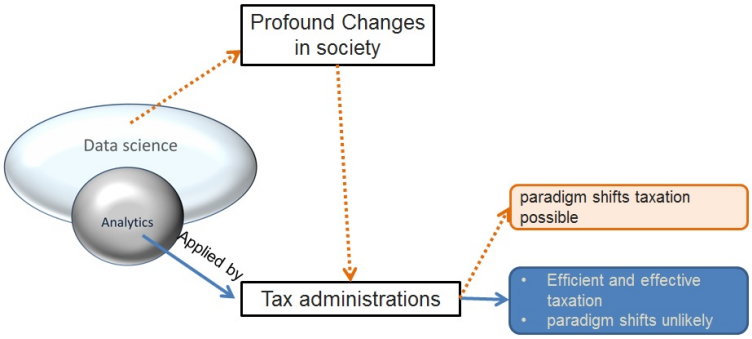


Figure 8.1: Two ways that data science influences a tax administration

8.2 Logistic Regression and Factorization Machines

The research question that was addressed in this chapter was:

> **Research Question** *Can we improve on current audit selection models by incorporating categorical variables with many values (like 'industry sector') that are valued by experts doing manual audit selection?*

The reason to address this question was the (at that time) surprising observation that in the manual selection of audits, categorical variables with many values were highly valued, while they were not incorporated in standard risk models at the NTCA.

After completing the research, it can be said that current audit selection models *can* be improved by including categorical variables with many values. The risk model that had started the research (a standard VAT audit selection model) could be improved by nearly 10%. We developed a technique that uses the recently developed Factorization Machines to achieve this goal.

The contribution is important beyond the actual problem at the NTCA. Many data scientists encounter categorical features with many levels and are looking for the right way to incorporate them in risk models or other applications. This is evident when looking for instance at the questions posed on this topic at Stack Overflow (www.stackoverflow.com), a main question and answer site in the domain of computer science. The solution described in the chapter, solves the problem in a satisfying way. The solution described transforms the categorical features in a data preprocessing step into numerical features. These numerical features can then be used in any classification or regression algorithm.

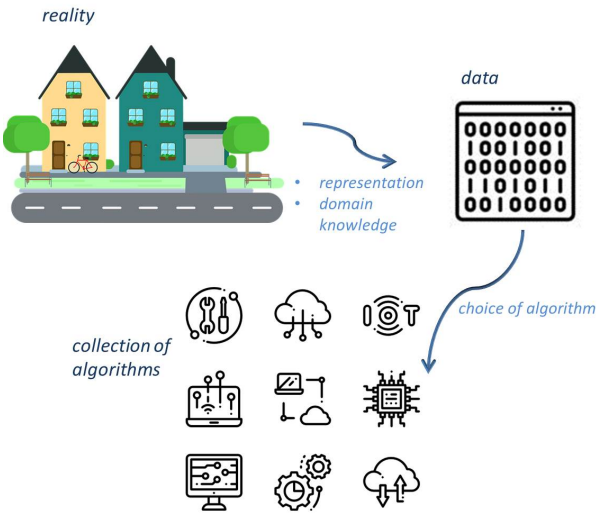


Figure 8.2: Visualization of the relation between the real world and data, stressing the importance of the way of representing the real world.

The idea of Factorization Machines to present categorical levels as vectors in a multidimensional space is attractive by itself: a multidimensional space contains a natural distance measure, allowing to quantify subtle differences between various levels. A spectacular example of this kind is presented by *word2vec* [101], a group

of models used in word embedding. These models are able to present each word as a vector, such that differences between vectors have an intrinsic meaning, like the famous example of *king - man + woman = queen*, where *king* is the vector belonging to the word 'king', *man* the vector belonging to the word 'man', et cetera. The plus and the minus sign in the equation are the usual addition and subtraction of vectors.

The representation of (interactions of) categorical features as numeric vectors is an example of an unconventional way of presenting reality in data. In this thesis, we met another example of the importance of a good data representation when discretizing numerical data to make it suitable as input for a Restricted Boltzmann Machine, see 'thermometer encoding' in Chapter 6 on page 83. Figure 8.2 shows that in general the representation of reality as input for an algorithm is an essential intermediate step. Data representation is often given little attention, but is important. We expect more good results in many application areas if more attention is paid to the representation of the data.

8.3 Singular Outliers

When applying standard anomaly detection algorithms to find tax fraud, we noticed that most of the anomalies found were not related to fraud. This sparked the following research question:

> **Research Question** *Can we develop a new anomaly detection algorithm tailored to find the outliers that are of interest to a tax administration?*

The answer is affirmative. In Chapter 4 we argue that interesting outliers do not necessarily have anomalous values for all features. Observations with anomalous values for one or a few features might be even of more interest in the domain of tax fraud detection or detecting data quality issues. The phenomenon of singular outliers is not only restricted to the tax domain. Also in other fraud types, like credit card fraud, clues are often derived from one or a few features.

The algorithm (SODA) that we developed is the first algorithm developed to detect this type of outliers. We successfully applied a previous version of the algorithm to select VAT field audits.

Since it is a first algorithm. we can imagine that other algorithms may be found in the future that can perform even better. For instance, existing anomaly detection algorithms may be adapted to find singular outliers. Also, the current algorithm may be refined by by taking other values for p instead of 2 and 1 in the Local Euclidean Manhattan Ratio, see Section 4.5.

At this place we want to point to a similarity between singular outlier detection and sub-group discovery. Sub-group discovery can be seen as clustering observations

on a subset of the features [8]. Before the arrival of sub-group discovery, many clustering techniques existed that clustered observations taking into account *all* features. Sub-group discovery became popular when it was realized that in some applications, interest lies in observations that form clusters when restricted to a few features. This is comparable with singular outlier detection where outliers are to be found for a few features.

8.4 Are Similar Cases Treated Similarly?

'Similar treatment of similar cases' is important for tax administrations. The phrase took a prominent place in the strategic five-year plan of the NTCA. Although it is easy to state this principle theoretically, it is not easy to impose it in practice in a large organization. Many factors may influence the treatment of taxpayers, especially for Knowledge Intensive processes, that are abundant at tax administrations. For this reason, we formulated our fifth research question as follows,

Research Question *Can we develop a data science technique that assists in testing similar treatment of similar (tax) cases?*

The test procedures introduced in Chapter 5, see pages 70 and 72, give a tool to test on similar treatment of similar cases. The procedures expect a process log as input, extract features and frequencies, and then apply a sequence of χ^2-tests. In case the Null hypothesis of similar treatment is rejected, the values of the individual χ^2-statistics allow to pinpoint the places in the process that most contributed to the rejection. The test procedures that are developed belong to *process mining*.

Process mining is still a relatively young field of research, but has a lot of potential for tax administrations and other organizations. Since processes are more and more controlled by state of the art case management systems that record valuable data, the tools of process mining can be used to optimize these processes. However, almost no tools exist yet for automatically checking *constraints* of processes, like the similar treatment of similar cases. We think that the use of statistical techniques, as is done in Chapter 5, can contribute further to process mining.

8.5 Extending an Anomaly Detection Benchmark

The topic of Chapter 6 is somewhat different compared to the other chapters. Its contribution lies in comparing the performance of existing algorithms. The comparison of algorithms contributes to the general body of knowledge, since science is not only about discovering new facts, but also about placing these facts in an agreed, coherent framework.

Comparing different algorithms also has practical value. A frequently asked question from data scientists is what algorithm should be used to solve a problem at hand. We can answer such questions, as soon as we know the advantages and disadvantages of various algorithms, and in what situations the strength of a particular algorithm can be utilized. The formal research question underlying chapter 6 is:

Research Questions *Unsupervised anomaly detection algorithms play an important role in (tax) fraud detection. What algorithms can be expected to work well under what conditions?*

The main finding of the chapter is that some exotic algorithms like Restricted Boltzmann Machines, or some new algorithms like Isolation Forests perform well on a range of data sets. For some data sets the investigated algorithms beat the current best algorithm. The results of the chapter are summarized in Table 6.3.

An interesting finding of the chapter is that simple algorithms for detecting anomalies, such as k-nearest neighbors and Histogram Based Outlier Score, perform well on a large number of data sets. This suggests a practical strategy to first try simple algorithms for a problem and then to build further from these results with more specific algorithms, like the ones tested in Chapter 6.

8.6 Outlook

Tax authorities around the world will continue to be large information processing organizations. If current trends of an increasingly dynamic society, increasing citizen expectations and internationalization continue, the tax administrations will face enough challenges. Fortunately, the constant developments in computer science can offer instruments to deal with an increasing workload. The application of data science in the tax authorities has started, but has not yet reached the highest maturity levels. In the near future, we expect progress in at least the following topics, that are of interest for other organizations as well,

- automatically categorizing incoming documents,
- developing new algorithms for detecting (tax) fraud in networks of agents,
- privacy-preserving data mining.

Certainly, not all challenges of data science and taxation are of a technical nature. When tax administrations are embracing data science, organizational and ethical questions arise, like,

- How to position data science in relation to classical ICT?

- How to organize teams of data scientist for optimal cooperation?
- How to organize a portfolio process?
- How to organize quality assurance and quality control?

The organizational questions may be of interest to Business Science.

Moreover ethical questions are playing an important role already, and will become even more important in the coming years. These are questions like,

- How to preserve the privacy of citizens and how to conform to privacy guidelines?
- In what cases do we allow (semi-) automatic decision making and in what cases do we want to avoid this?
- How can we develop effective controls that prevent the emergence of a police state?

The right for the protection of privacy is stated in various legal documents, like the Convention for the Protection of Human Rights and Fundamental Freedoms, article 8, see [28].

The question of automatic decision making is relevant since computer systems may miss important information about the context, and may therefore take a wrong decision. Moreover, it may be hard for a citizen to dispute such an automatic decision once human process workers have been replaced by cheaper, more efficient machines. Currently, the view is emerging that intelligent computer systems should support human decision making, and not take over complex decisions with a wide impact, see for instance the European Union Communication on Building Trust in Human-Centric Artificial Intelligence [35].

The power that data science and artificial intelligence may give to governments, is subject to debate currently [37, 84]. Although this power may do much good, it may be used by a future government to tightly control its citizens and to undermine democratic principles, like individual freedoms. Regulatory frameworks may be needed to prevent if one wants to avoid this.

Summary

This dissertation is about applications of data science techniques to the administration of taxes. These applications can provide effective and efficient processes within a tax administration or improve the compliance of taxpayers. On the one hand, existing techniques are assessed, and on the other, new techniques have been developed.

After an introductory chapter, the question is addressed which processes in the tax domain are suitable for improvement with data science applications (Chapter 2). Because the entire tax domain is extensive, we limit ourselves to an important subdomain: *taxpayer supervision*. A subtle picture of the (im-)possibilities of data science is created by making an inventory of the supervisory activities and investigating for each activity which data science techniques can add value. The chapter also provides an overview of techniques that can be deployed. As such it is applicable for tax authorities that want to improve their supervisory activities with data science.

In Chapter 3 an existing data science technique, *logistic regression*, is extended. Logistic regression has difficulty in getting information efficiently from certain types of data, the so-called 'categorical variables with many levels'. Examples are 'place of residence' or 'occupation'. Because these variables occur regularly with tax authorities and because logistic regression is often used as well, this extension is a useful addition. Categorical variables with many levels occur also outside the domain of taxation, so this extension may add value there as well. Probably the same techniques of incorporating categorical variables with many levels can also be applied to techniques other than logistic regression, but this has not been investigated further.

In Chapter 4 we look at anomaly detection techniques (i.e. finding outliers in data). Tax data often includes a typical type of anomalies (also called *outliers*) that are not well detected by existing techniques. In the chapter this type of outliers is described and a new technique has been developed that finds these anomalies. Within tax administrations, the technique makes sense to detect tax fraud or to improve the

data quality by detecting errors in data entry.

Chapter 5 is about similar treatment of similar cases. For large organizations, such as tax authorities, it is not easy to ensure that equal cases always receive equal treatment. In particular for those processes that require a human assessment ('Knowledge Intensive Processes'). A first step to arrive at similar treatment is to *test* whether similar cases are treated similarly and, if not, where in the process the dissimilar treatment occurs. For this purpose, two new procedures are introduced in Chapter 5. These procedures belong to the domain of *process mining*.

Science consists of discovering new knowledge, but also organizing that knowledge in logical frameworks. In Chapter 6 the latter topic is looked at: three little-used / new anomaly detection techniques are being tested on a benchmark, recently published by Goldstein and Uchida [43]. This comparison helps in understanding the strengths and weaknesses of various anomaly detection techniques. Surprisingly, the three new techniques are able to match or improve the benchmark for 30% of the data sets. In addition, attention is given to the relation between the new methods and the more traditional anomaly detection methods. Moreover we look at what type of anomalies are found.

In Chapter 7 four smaller topics are discussed. The chapter gives a good impression of the variety of data science topics within the tax domain. The first subsection deals with a topic from HR Analytics, a sub-area within data science that focuses on applications within the Human Resources (HR) domain. The next subsection looks at an application of Reinforcement Learning techniques within the collection process of a tax authority. Reinforcement Learning is the branch of Artificial Intelligence that focuses on learning strategies for board games and computer games. The third topic is about explaining data science models to end-users. The last topic concerns applying ideas from the Fuzzy Set community in tax domain.

The dissertation ends with conclusions and a short outlook.

Samenvatting

Datawetenschap voor Belastingdiensten

Dit proefschrift gaat over toepassingen van technieken uit de datawetenschappen ('data science') voor het verbeteren van heffings- en inningsprocessen binnen belastingdiensten. De toepassing van data science technieken kan leiden tot effectievere en/of efficiëntere processen of de compliantie (d.w.z. het vrijwillig naleven van regelgeving) bij burgers en bedrijven verhogen. Enerzijds worden bestaande technieken beoordeeld op hun toepasbaarheid in dit domein, anderzijds zijn nieuwe technieken ontwikkeld.

Na een inleidend hoofdstuk, staat in Hoofdstuk 2 de vraag centraal welke fiscale processen zich lenen voor verbetering door middel van data science technieken. Omdat het hele belastingdomein omvangrijk is, beperken we ons tot een belangrijk deelgebied: *het toezicht*. Door de toezichtsactiviteiten te categoriseren en voor elke categorie actviteiten na te gaan welke data science technieken waarde kunnen toevoegen, ontstaat een genuanceerd beeld van de (on-)mogelijkheden van data science. Het hoofdstuk geeft ook een overzicht van technieken die ingezet kunnen worden en is als zodanig bruikbaar voor belastingdiensten die verbeteringen willen realiseren in hun toezichtsactiviteiten met behulp van data science.

In Hoofdstuk 3 wordt een bestaande data science techniek, *logistische regressie*, uitgebreid. Logistische regressie heeft namelijk moeite met een bepaald type gegevens, de zogenaamde 'categoriale variabelen met veel waarden'. Denk bijvoorbeeld aan gegevens als 'woonplaats' of 'beroep'. Omdat deze variabelen regelmatig voorkomen bij belastingdiensten en omdat ook logistische regressie vaak wordt ingezet, is deze uitbreiding een nuttige aanvulling. Ook buiten het domein van belastingen komen categoriale variabelen met veel waarden voor en ook daar kan deze bijdrage nuttig zijn. Waarschijnlijk zijn de technieken om categoriale variabelen in te bedden

ook toe te passen bij andere voorspellende modellen dan logistische regressie, maar dit is niet verder onderzocht.

In Hoofdstuk 4 wordt gekeken naar anomalie detectie technieken (d.w.z. technieken om uitschieters in de data te vinden). Binnen belastingdata komen vaak een typische soort anomalieën voor die niet zo goed wordt gedetecteerd door bestaande anomalie technieken. In het hoofdstuk worden deze soort uitschieters beschreven en is er een nieuwe techniek ontwikkeld die deze anomalieën vindt. Binnen belastingdiensten is de techniek zinvol om in te zetten om belastingfraude te ontdekken of voor fouten in (handmatige) data-invoer op te sporen.

In Hoofdstuk 5 wordt ingegaan op een principe van rechtsgelijkheid: 'vergelijkbare gevallen dienen vergelijkbaar behandeld te worden'. Voor processen binnen een grote organisatie als een belastingdienst is het niet eenvoudig om te bewerkstelligen dat vergelijkbare gevallen een vergelijkbare behandeling krijgen, met name voor die processen die een menselijke beoordeling bevatten. Een eerste stap om te komen tot een vergelijkbare behandeling is het *meten* of gevallen vergelijkbaar behandeld worden en zo nee, waar in het proces deze ongelijkheid voorkomt. Hiervoor zijn in Hoofdstuk 5 twee nieuwe procedures ontwikkeld en beschreven. Deze nieuwe technieken behoren bij het vakgebied *process mining*.

In Hoofdstuk 6 wordt een ander aspect van de informatica-wetenschap opgepakt. Wetenschap bestaat niet enkel uit het ontdekken van nieuwe kennis, maar ook het ordenen van die kennis in logische categorieën. In dit hoofdstuk worden drie weinig gebruikte / nieuwe anomalie detectie technieken getest op een benchmark, recentelijk gepubliceerd door Goldstein en Uchida [43]. De vergelijking helpt om de sterke en zwakke punten van de verscheidene technieken te begrijpen. Verrassend blijkt dat de drie nieuwe technieken in 30% van de datasets in staat zijn om de benchmark te evenaren of te verbeteren. Daarnaast is het interessant om te zien hoe de nieuwe methoden zich verhouden tot de meer klassieke anomalie detectie methoden en welk soort anomalieën gevonden worden.

In Hoofdstuk 7 worden een viertal kleinere onderwerpen besproken. Dit hoofdstuk geeft een indruk van de verscheidenheid van data science toepassingen binnen het belastingdomein. De eerste paragraaf behandelt een onderwerp uit *HR Analytics*, een deelgebied binnen data science dat zich richt op toepassingen binnen Human Resources (HR). In de volgende paragraaf komt een toepassing van Reinforcement Learning technieken aan bod binnen het inningsproces van belastingdiensten. Reinforcement Learning is de tak van Artificiële Intelligentie die zich bezighoudt met het leren van strategieën bij bordspelen en computer games. Het derde onderwerp gaat over het uitleggen van risicomodellen aan eindgebruikers. Het laatste onderwerp betreft het toepassen van ideeën uit de Fuzzy Set gemeenschap op de belastingheffing.

Het proefschrift wordt afgesloten met conclusies en een korte vooruitblik.

Acknowledgements

I am grateful for the many great moments this PhD track has brought to me. It offered an opportunity to see the richness of the field and I have been able to meet many fascinating people. In particular I would like to thank the persons below.

To start, I would like to thank Annette for encouraging me to follow the direction of my heart. Without her support, I would not have dared to start this PhD track. Also, I would like to thank Gert Willem Haasnoot from The Curious Network and Joan van der Zee, coach at the Belastingdienst. Both have encouraged me at the initial stage when the PhD track was still just an idea. With hindsight, the decision to quit my position as a CFO advisor and to start working as a data scientist and part-time PhD candidate has been a good one.

Many thanks also to the people that made possible this track, in particular Nicole Back and Cees Bredius, directors at the NTCA, who put confidence in the track and me. I also appreciate dearly the possibilities the NTCA has offered to complete this piece of research. I am also grateful for the help that was given by prof. dr. mr. Lisette van der Hel RA and prof. dr. Bert Kersten, to initialize the PhD track.

All beginnings are difficult. In the first year, Herman Hoorweg has played a pivotal role. His curiosity to know what Analytics can actually contribute to a tax administration has been the spark for the first article. After that initial paper, spotting interesting topics has been much easier.

The greatest 'thank you' goes without any doubt to my co-promotor dr. Wojtek Kowalczyk. I want to thank him for being a great master in Data Science for me. I also want to thank him for the courage to start this PhD in the first place: a track that has been surrounded by many uncertainties initially. And last but not least, for being a wonderful person. I will definitely miss our talks.

Finally, I would like to thank dr. Frank Takes for providing the latex template for this dissertation.

Curriculum Vitae English

Mark Pijnenburg was born on May 7, 1979 in Heerlen. Mark completed grammar school at the Bernardinuscollege in 1997 (summa cum laude). In 2003 he obtained his master's degree in Mathematics (cum laude) from the Radboud University.

After this, Mark worked for two years for Eindhoven University of Technology in the program 'Mathematics for Industry' of the Stan Ackermans Institute. After a final project at DSM Research, the program was completed in 2005. In 2009, the Amsterdam MBA was completed at the University of Amsterdam (cum laude).

In 2006 Mark joined the Tax Authorities in Utrecht. Various positions followed with the Tax Authorities, including 'quantitative researcher', 'business analyst' and 'advisor to the CFO'. In 2014, the position started as a Data Scientist. And later that year, the parallel PhD track started at the Leiden Institute for Advanced Computer Science (LIACS) of Leiden University, with this thesis as the final result.

Curriculum Vitae Nederlands

Mark Pijnenburg is geboren op 7 mei 1979 te Heerlen. In 1997 rondde Mark het gymnasium af bij het Bernardinuscollege (summa cum laude). In 2003 behaalde hij aan de Radboud Universiteit zijn master Wiskunde (cum laude).

Hierna werkte Mark twee jaar voor de Technische Universiteit Eindhoven in het programma 'Mathematics for Industry' van het Stan Ackermans-instituut. Na een eindopdracht bij DSM Research werd het programma in 2005 afgerond. In 2009 werd de Amsterdam MBA afgerond bij de Universiteit van Amsterdam (cum laude).

In 2006 trad Mark in dienst bij de Belastingdienst te Utrecht. Bij de Belastingdienst volgden verschillende functies, waaronder kwantitatief onderzoeker, business analist en adviseur van de Chief Financial Officer. In 2014 startte de functie als Data Scientist. En later dat jaar begon het parallelle promotietraject bij het Leiden Institute for Advanced Computer Science (LIACS) van de Universiteit Leiden, met dit proefschrift als eindresultaat.

Bibliography

[1] N. G. C. F. M. Abreu et al. *Analise do perfil do cliente Recheio e desenvolvimento de um sistema promocional*. PhD thesis, ISCTE-IUL, 2011.

[2] L. Acebedo and J. Durnall. Risk-based pricing: When does it work and when does it not? an adverse selection approach. Technical report, Experian Decision Analytics, 5 2013.

[3] O. Akbilgic, H. Bozdogan, and M. E. Balaban. A novel hybrid rbf neural networks model as a forecaster. *Statistics and Computing*, 24(3):365–375, 2014.

[4] S. Akter and S. F. Wamba. Big data analytics in e-commerce: a systematic review and agenda for future research. *Electronic Markets*, 26(2):173–194, 2016.

[5] M. G. Allingham and A. Sandmo. Income tax evasion: A theoretical analysis. *Journal of public economics*, 1(3-4):323–338, 1972.

[6] J. Alm. Measuring, explaining, and controlling tax evasion: lessons from theory, experiments, and field studies. *International tax and public finance*, 19(1):54–77, 2012.

[7] J. Andreoni, B. Erard, and J. Feinstein. Tax compliance. *Journal of economic literature*, 36(2):818–860, 1998.

[8] M. Atzmueller. Subgroup discovery. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 5(1):35–49, 2015.

[9] R. Bakeman and J. M. Gottman. *Observing interaction: An introduction to sequential analysis*. Cambridge university press, 1997.

[10] D. Bassi and C. Hernandez. Credit risk scoring: results of different network structures, preprocessing and self-organised clustering. In *Decision Technologies for Financial Engineering. Proceedings of the Fourth International Conference on Neural Networks in the Capital Markets*, pages 151–61, 1997.

[11] S. Basta, F. Fassetti, M. Guarascio, G. Manco, F. Giannotti, D. Pedreschi, L. Spinsanti, G. Papi, and S. Pisani. High quality true-positive prediction for fiscal fraud detection. In *Data Mining Workshops, 2009. ICDMW'09. IEEE International Conference on Data Mining*, pages 7–12. IEEE, 2009.

[12] S. Basu and G. B. Waymire. Recordkeeping and human evolution. *Accounting Horizons*, 20(3):201–229, 2006.

[13] Overzicht verwerkingen van persoonsgegevens door de Belastingdienst. `https://download.belastingdienst.nl/belastingdienst/docs/verw_persoonsgegevens_belastingdienst_al5303z7fd.pdf`. Accessed: 2019-07-01.

[14] Belastingdienst. Claiming refunds for vat. `https://www.belastingdienst.nl/wps/wcm/connect/bldcontenten/belastingdienst/business/vat/vat_in_the_netherlands/claiming_refund_of_vat/claiming_refund_of_vat`, 2019.

[15] V. Bentkus, G. Geuze, M. Pijnenburg, and M. van Zuijlen. Unimodality: The symmetric case. Technical report, Radboud University Nijmegen, 2006.

[16] N. C. Berkman. Value grouping for binary decision trees. Technical report, University of Massachusetts, 1995.

[17] A. Bolt, W. van der Aalst, and M. de Leoni. Finding process variants in event logs (short paper). In *Confederated International Conference On the Move to Meaningful Internet Systems, OTM 2017 held in conjunction with Conferences on CoopIS, CandTC and ODBASE 2017*. Springer Netherlands, 2017.

[18] C. Boon, F. D. Belschak, D. N. Den Hartog, and M. Pijnenburg. Perceived human resource management practices. *Journal of Personnel Psychology*, 2014.

[19] R. J. C. Bose, W. M. Van Der Aalst, I. Zliobaite, and M. Pechenizkiy. Dealing with concept drifts in process mining. *IEEE transactions on neural networks and learning systems*, 25(1):154–171, 2014.

[20] V. Braithwaite. Responsive regulation and taxation: Introduction. *Law & Policy*, 29(1):3–10, 2007.

[21] L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and regression trees*. CRC press, 1984.

[22] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: identifying density-based local outliers. In *ACM sigmod record*, volume 29, pages 93–104. ACM, 2000.

[23] C. Brunsdon, A. Fotheringham, and M. Charlton. An investigation of methods for visualising highly multivariate datasets. *Case Studies of Visualization in the Social Sciences*, pages 55–80, 1998.

[24] D. Burshtein, V. Della Pietra, D. Kanevsky, and A. Nadas. Minimum impurity partitions. *The Annals of Statistics*, pages 1637–1646, 1992.

[25] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[26] P. A. Chou et al. Optimal partitioning for classification and regression trees. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):340–354, 1991.

[27] R. B. Cialdini and N. J. Goldstein. Social influence: Compliance and conformity. *Annu. Rev. Psychol.*, 55:591–621, 2004.

[28] Council of European Union. Convention for the protection of human rights and fundamental freedoms. `https://www.coe.int/en/web/conventions/full-list/-/conventions/treaty/005`, 1953.

[29] D. Cramer. *Advanced quantitative data analysis*. McGraw-Hill Education (UK), 2003.

[30] L. S. da Silva, R. N. Carvalho, and J. C. F. Souza. Predictive models on tax refund claims-essays of data mining in brazilian tax administration. In *International Conference on Electronic Government and the Information Systems Perspective (EGOVIS)*, pages 220–228. Springer, 2015.

[31] T. Davenport. *Big data at work: dispelling the myths, uncovering the opportunities*. Harvard Business Review Press, 2014.

[32] T. Davenport and J. Harris. Competing on analytics. *Boston, MA: Harvard Business School Publishing*, 2007.

[33] T. H. Davenport, J. G. Harris, and R. Morison. *Analytics at work: Smarter decisions, better results*. Harvard Business Press, 2010.

[34] E. Dubossarsky and Y. Tyshetskiy. R package autoencoder. `https://CRAN.R-project.org/package=autoencoder`, May 2014.

[35] European Commission. Building trust in human centric artificial intelligence. `https://ec.europa.eu/digital-single-market/en/news/communication-building-trust-human-centric-artificial-intelligence`, 4 2019.

[36] W. Federer. *Statistics and Society*. Marcel Dekker Inc., 2 edition, 1991.

[37] S. Feldstein. The road to digital unfreedom: How artificial intelligence is reshaping repression. *Journal of Democracy*, 30(1):40–52, 2019.

[38] Fiscalis Risk Management Platform Group. Compliance risk management guide for tax administrations. `http://ec.europa.eu/taxation_customs/resources/documents/©common/publications/info_docs/taxation/risk_managt_guide_en.pdf`, 5 2010.

[39] A. Fischer and C. Igel. An introduction to restricted boltzmann machines. In *iberoamerican congress on pattern recognition*, pages 14–36. Springer, 2012.

[40] Forum on Tax Administration. Advanced analytics for better tax administration: Putting data to work. `http://dx.doi.org/10.1787/9789264256453-en`, 2016.

[41] Forum on Tax Administration Compliance Sub-group Compliance Risk Management. Managing and improving tax compliance, guidance note. `http://www.oecd.org/tax/administration/33818656.pdf`, 2004.

[42] J. Friedman, T. Hastie, and R. Tibshirani. The elements of statistical learning: Data mining, inference, and prediction. *Springer Series in Statistics (*, 2009.

[43] M. Goldstein and S. Uchida. A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data. *PloS one*, 11(4):e0152173, 2016.

[44] G. Gupta. *Introduction to data mining with case studies*. PHI Learning Pvt. Ltd., 2014.

[45] M. Gupta and V. Nagadevara. Audit selection strategy for improving tax compliance–application of data mining techniques. In *Foundations of*

Risk-Based Audits. Proceedings of the eleventh International Conference on e-Governance, Hyderabad, India, December, pages 28–30, 2007.

[46] H. Haddadi, R. Mortier, and S. Hand. Privacy analytics. *ACM SIGCOMM Computer Communication Review*, 42(2):94–98, 2012.

[47] A. Hald. Statistical theory with engineering applications. In *Statistical theory with engineering applications*. John Wiley & Sons, 1952.

[48] S. Hariri, M. C. Kind, and R. J. Brunner. Extended isolation forest. *arXiv preprint arXiv:1811.02141*, 2018.

[49] K. Hassibi et al. Detecting payment card fraud with neural networks. *World Scientific Book Chapters*, pages 141–157, 2000.

[50] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction, Springer Series in Statistics*. Springer New York, 2 edition, 2009.

[51] G. E. Hinton. A practical guide to training restricted boltzmann machines. In *Neural networks: Tricks of the trade*, pages 599–619. Springer, 2012.

[52] G. E. Hinton and R. R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[53] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[54] K. Hsu, N. Pathak, J. Srivastava, G. Tschida, and E. Bjorklund. Data mining based tax audit selection: A case study from minnesota department of revenue. In *Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2009.

[55] B. R. Jackson and V. C. Milliron. Tax compliance research: Findings, problems, and prospects. *Journal of accounting literature*, 5(1):125–165, 1986.

[56] G. K. Kanji. *100 Statistical Tests*. Sage Publications, 3 edition, 2006.

[57] G. V. Kass. An exploratory technique for investigating large quantities of categorical data. *Applied statistics*, pages 119–127, 1980.

[58] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, Aug. 2009.

[59] A. Kumar, A. Saxena, V. Suvarna, and V. Rawat. Top 10 trends in banking in 2016. `https://www.capgemini.com/wp-content/uploads/2017/07/banking_top_10_trends_2016.pdf`, 3 2016.

[60] D. Kwon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim. A survey of deep learning-based network anomaly detection. *Cluster Computing*, pages 1–13, 2017.

[61] D. Larose. *Discovering Knowledge in Data*. New Jersey: John Willey & Sons, 2005.

[62] K. C. Laudon, C. G. Traver, et al. *E-commerce: business, technology, society*. Pearson, 2016.

[63] J. Leskovec, A. Rajaraman, and J. D. Ullman. *Mining of massive datasets*. Cambridge university press, 2 edition, 2014.

[64] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.

[65] M. Lichman. UCI machine learning repository. `http://archive.ics.uci.edu/ml`, 2013.

[66] R. Lieber. Lower your car insurance bill, at the price of some privacy. `www.nytimes.com/2014/08/16/your-money/auto-insurance/tracking-gadgets-could-lower-your-car-insurance-at-the-price-of-some-privac` `html`, 2014.

[67] G. S. Linoff and M. J. Berry. *Data mining techniques: for marketing, sales, and customer relationship management*. John Wiley & Sons, 3 edition, 2011.

[68] Z. C. Lipton, J. Berkowitz, and C. Elkan. A critical review of recurrent neural networks for sequence learning. *arXiv preprint arXiv:1506.00019*, 2015.

[69] B. Liu. *Web data mining: exploring hyperlinks, contents, and usage data*. Springer Science & Business Media, 2007.

[70] F. T. Liu. R package isolationforest. `https://rdrr.io/rforge/IsolationForest/`, August 2009.

[71] F. T. Liu, K. M. Ting, and Z.-H. Zhou. Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422. IEEE, 2008.

[72] A. Maaradji, M. Dumas, M. La Rosa, and A. Ostovar. Fast and accurate business process drift detection. In *International Conference on Business Process Management*, pages 406–422. Springer, 2016.

[73] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal. Long short term memory networks for anomaly detection in time series. In *Proceedings*, page 89. Presses universitaires de Louvain, 2015.

[74] K. Mardia, J. Kent, and J. Bibby. Multivariate statistics, 1979.

[75] M. A. Marin, M. Hauder, and F. Matthes. Case management: an evaluation of existing approaches for knowledge-intensive processes. In *International Conference on Business Process Management*, pages 5–16. Springer, 2015.

[76] J. Meza, L. Terán, A. Piaún, and M. Tomalá. A fuzzy-based recommender system for public tax payment. In *2018 International Conference on eDemocracy & eGovernment (ICEDEG)*, pages 235–240. IEEE, 2018.

[77] D. Micci-Barreca. A preprocessing scheme for high-cardinality categorical attributes in classification and prediction problems. *ACM SIGKDD Explorations Newsletter*, 3(1):27–32, 2001.

[78] K. P. Murphy. *Machine Learning A Probabilistic Perspective*. The MIT Press, 1 edition, 2012.

[79] Netherlands Tax and Customs Administration. `download.belastingdienst.nl/belastingdienst/docs/dutch_tax_customs_admin.pdf`, May 2019.

[80] A. Ng. Lecture notes on sparse autoencoders. `https://web.stanford.edu/class/cs294a/sparseAutoencoder-2011.pdf`, 2011.

[81] Online etymology dictionary, statistics. `https://www.etymonline.com/word/statistics`.

[82] A. Ostovar, S. J. Leemans, and M. La Rosa. Robust drift characterization from event streams of business processes. *Internal Report*, 2018.

[83] S. Pauwels and T. Calders. Detecting and explaining drifts in yearly grant applications. *arXiv preprint arXiv:1809.05650*, 2018.

[84] N. Petit. Artificial intelligence and automated law enforcement: A review paper. *Available at SSRN 3145133*, 2018.

[85] M. Pijnenburg. Simulation of rubber networks. Technical report, Stan Ackermans instituut, 2005. ISBN: 9044404784.

[86] M. Pijnenburg. Code used in experiments. `https://github.com/PijnenburgMark/anomaly_detection_benchmark`, 2019. Accessed: 2019-06-01.

[87] M. Pijnenburg, N. Kalosha, and M. C. van Zuijlen. Hoeffding-Bentkus bound in statistical auditing. Technical report, Radboud University Nijmegen, 2006.

[88] M. Pijnenburg and W. Kowalczyk. Applying analytics for improved taxpayer supervision. In *Proceedings of 16th European Conference on e-Government ECEG 2016*, pages 145–153. Academic Conferences and publishing limited, 2016.

[89] M. Pijnenburg and W. Kowalczyk. Extending logistic regression models with factorization machines. In *International Symposium on Methodologies for Intelligent Systems*, pages 323–332. Springer, 2017.

[90] M. Pijnenburg and W. Kowalczyk. Singular outliers: Finding common observations with an uncommon feature. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 492–503. Springer, 2018.

[91] M. Pijnenburg and W. Kowalczyk. Are similar cases treated similarly? a comparison between process workers. In *International Conference on Business Information Systems*, pages 1–15. Springer, 2019.

[92] M. Pijnenburg and W. Kowalczyk. Extending an anomaly detection benchmark with auto-encoders, isolation forests, and rbms. In *International Conference on Information and Software Technologies*. Springer, 2019. Best Paper Award.

[93] M. Pijnenburg, W. Kowalczyk, E. van der Hel-van Dijk, et al. A roadmap for analytics in taxpayer supervision. *Electronic Journal of e-Government*, 15:19–32, 2017.

[94] M. Pijnenburg and K. Kuijpers. Explaining risk models to the business. *Tax Tribune*, 35:57–62, 2016.

[95] M. Post. Tax data and reinforcement learning. Master's thesis, Leiden University, 2 2019. Under supervision of Mark Pijnenburg, Wojtek Kowalczyk, and Kaifeng Yang.

[96] S. Ramaswamy, R. Rastogi, and K. Shim. Efficient algorithms for mining outliers from large data sets. In *ACM Sigmod Record*, volume 29, pages 427–438. ACM, 2000.

[97] S. Rendle. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[98] S. Rendle. Factorization machines with libFM. *ACM Trans. Intell. Syst. Technol.*, 3(3):57:1–57:22, May 2012.

[99] M. Richardson and A. J. Sawyer. A taxonomy of the tax compliance literature: further findings, problems and prospects. *Austl. Tax F.*, 16:137–320, 2001.

[100] X. Rong. R package deepnet. `https://CRAN.R-project.org/package=deepnet`, March 2014.

[101] X. Rong. word2vec parameter learning explained. *arXiv preprint arXiv:1411.2738*, 2014.

[102] M. Sabokrou, M. Fayyaz, M. Fathy, Z. Moayed, and R. Klette. Deep-anomaly: Fully convolutional neural network for fast anomaly detection in crowded scenes. *Computer Vision and Image Understanding*, 172:88–97, 2018.

[103] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In *International Conference on Information Processing in Medical Imaging*, pages 146–157. Springer, 2017.

[104] H. Shane. Insurance cheats discover social media is the real pain in the neck. `www.theguardian.com/money/2016/jul/18/insurance-cheats-social-media-whiplash-false-claimants`, 2016.

[105] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 1 edition, 11 2017. complete draft.

[106] United nations of roma victrix (unrv), roman taxes. `https://www.unrv.com/economy/roman-taxes.php`.

[107] L. van der Maaten, E. Postma, and H. van den Jerik. Dimensionality reduction: A comparative review. Technical report, Maastricht University, 2009.

[108] F. van Dongen, B.F.; Borchert. Bpi challenge 2018. `https://doi.org/10.4121/uuid:3301445f-95e8-4ff0-98a4-901f1f204972`, 2018.

[109] S. F. Wamba and S. Akter. Big data analytics for supply chain management: A literature review and research agenda. In *Workshop on Enterprise and Organizational Modeling and Simulation*, pages 61–72. Springer, 2015.

[110] E. Wiebes. Investeringsagenda belastingdienst. `https://www.tweedekamer.nl/kamerstukken/detail?id=2015Z09033&did=2015D18368`, 2015.

[111] Wikipedia, English edition, history of statistics. `https://en.wikipedia.org/wiki/History_of_statistics#Etymology`.

[112] Wikipedia, English edition, trial of the pyx. `https://en.wikipedia.org/wiki/Trial_of_the_Pyx`.

[113] R.-S. Wu, C.-S. Ou, H.-y. Lin, S.-I. Chang, and D. C. Yen. Using data mining technique to enhance tax evasion detection performance. *Expert Systems with Applications*, 39(10):8769–8777, 2012.

[114] J. Xu, M. Saebi, B. Ribeiro, L. M. Kaplan, and N. V. Chawla. Detecting anomalies in sequential data with higher-order networks. *arXiv preprint arXiv:1712.09658*, 2017.

[115] D. Yates, D. Moore, and G. McCabe. *The Practice of Statistics*. W.H. Freeman, New York, 1 edition, 1999.

[116] M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. *Machine learning*, 42(1-2):31–60, 2001.

[117] M. Zikeba, S. K. Tomczak, and J. M. Tomczak. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Systems with Applications*, 2016.

Publication List Author

Below follows a chronological list of publications by the author.

1. M. Pijnenburg. Simulation of rubber networks. Technical report, Stan Ackermans instituut, 2005. ISBN: 9044404784

2. V. Bentkus, G. Geuze, M. Pijnenburg, and M. van Zuijlen. Unimodality: The symmetric case. Technical report, Radboud University Nijmegen, 2006

3. M. Pijnenburg, N. Kalosha, and M. C. van Zuijlen. Hoeffding-Bentkus bound in statistical auditing. Technical report, Radboud University Nijmegen, 2006

4. C. Boon, F. D. Belschak, D. N. Den Hartog, and M. Pijnenburg. Perceived human resource management practices. *Journal of Personnel Psychology*, 2014

5. M. Pijnenburg and W. Kowalczyk. Applying analytics for improved taxpayer supervision. In *Proceedings of 16th European Conference on e-Government ECEG 2016*, pages 145–153. Academic Conferences and publishing limited, 2016

6. M. Pijnenburg and K. Kuijpers. Explaining risk models to the business. *Tax Tribune*, 35:57–62, 2016

7. M. Pijnenburg, W. Kowalczyk, E. van der Hel-van Dijk, et al. A roadmap for analytics in taxpayer supervision. *Electronic Journal of e-Government*, 15:19–32, 2017

8. M. Pijnenburg and W. Kowalczyk. Extending logistic regression models with factorization machines. In *International Symposium on Methodologies for Intelligent Systems*, pages 323–332. Springer, 2017

9. M. Pijnenburg and W. Kowalczyk. Singular outliers: Finding common observations with an uncommon feature. In *International Conference on Information*

Processing and Management of Uncertainty in Knowledge-Based Systems, pages 492–503. Springer, 2018

10. M. Pijnenburg and W. Kowalczyk. Are similar cases treated similarly? a comparison between process workers. In *International Conference on Business Information Systems*, pages 1–15. Springer, 2019

11. M. Pijnenburg and W. Kowalczyk. Extending an anomaly detection benchmark with auto-encoders, isolation forests, and rbms. In *International Conference on Information and Software Technologies*. Springer, 2019. Best Paper Award