An Application of Business Rule Optimisation

Alan Roy Dormer

BSc. MSc. Tech

A thesis submitted for the degree of Doctor of Philosophy at

Monash University in 2020

Faculty of Information Technology

Copyright notice

Abstract

Business rules are intended to assert business structure or to control or influence the behaviour of a business. They are used widely throughout the services sector. They ensure compliance with statutory requirements and company policies, and provide consistency, allowing complex transactions to be processed automatically or by guiding relatively unskilled staff.

The focus of this thesis is a class of business rules categorised as computation, reasoning, and allocation (CRA) rules (Packt, 2009) as applied to a set of service business problems with common characteristics. These rules act on situational information to calculate additional quantities, process multiple inputs, and determine the path of the customer through a process. The rules are not mandatory, and there is scope for a business to change them to improve outcomes such as profit and customer service. These types of tasks have traditionally required human decision makers, but more commonly, business rules are used to automate and deskill the process.

While there is a large body of research on the construction, deployment, and operation of business rules, there are some areas that have not been addressed. Firstly, business rules can have an impact on the financial performance of an organisation. Secondly, merely replacing a human decision maker with a set of rules executed automatically may not always be the best option. Thirdly, rather than designing rules that emulate the human decision maker (and all their biases and faults), it may be better to build the rules to achieve the best outcome for the organisation. And finally, information is not necessarily free, and there is a need to balance the cost of gaining and processing information against its value in terms of better decisions.

This thesis considers a range of service business processes that result in a decision and applies business rule optimisation to a classical case study, that of credit approval, using real data. The application considers all relevant factors such as the potential profit, examination costs, the accuracy of the human expert and the rule system, and transaction abandonment. The case study demonstrates that business rule optimisation can be practical, useful, and beneficial in this context.

The contributions of this thesis include a practical method to build and optimise business rules that include human experts in the decision-making business

process. This research has developed theory and a framework that can be applied to create business rules in a recognisable format that can be understood by the user. The framework also includes a model of transaction abandonment and incorporates this into the feature selection and optimisation process. Finally, it enables the process to request further information, if required, and adjusts the rules to maintain maximum efficiency should the case-mix or caseload change.

Declaration

This thesis is an original work of my research and contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Publications during enrolment

Dormer, A. (2018). Business Rule Optimisation: Problem Definition, Proof-of-Concept and Application Areas. In International Conference on Business Information Systems (pp. 51-62). Springer.

Dormer, A. (2018). Cyborganisation: Machines and Humans Make Optimal Decisions Together. In Third International Congress on Information and Communication Technology: ICICT 2018, London (Vol. 797, pp. 487-497). Springer.

Dormer, A. (2017). A Framework for Optimising Business Rules. In International Conference on Business Information Systems (pp. 5-17). Springer.

Dormer, A. (2012). Optimising business rules in the services sector. Int. J. Soc. Behav. Educ. Econ. Bus. Ind. Eng, 6(10), 2580-2584.

CONTENTS

List of Tables

List of Figures

LIST OF ABBREVIATIONS AND ACRONYMS

BPM – Business Process Modelling

BPO – Business Process Optimisation

BRO – Business Rule Optimisation

Case – The subject of the business rules (potential customer, patient, etc.)

CRA – Computation, Reasoning and Allocation Rules

DTI – Debt to Income ratio

FICO – Credit score produced by Fair Isaac and Company

LIST OF APPENDICES

1 INTRODUCTION

Business rules are widely used in the services sector (Gartner, 2012). They provide consistency and allow relatively unskilled staff to process complex transactions correctly. But there are many examples where the rules themselves have an impact on the costs and profits of an organisation. Financial services, transport, and human services are areas where the rules themselves can predictably impact the bottom line. This situation creates an opportunity to find methods that can build a set of rules (or sets of rules) that will maximise profit, performance, or customer service, or any other key performance indicators. The manufacturing, energy and process industries have embraced mathematical optimisation techniques to improve efficiency, increase production, and reduce costs (Tiwari et al., 2015).

1.1 Importance of Business Rules

A business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business (Hay et al., 2000). Business rules in the services sector are about guiding or making decisions. As information is processed, decisions are made, and further information is generated. For example, a customer enters a bank and asks for a loan. Information is gathered about the customer and what they want, and the result of analysing that information other information is generated, e.g., the customer is accepted for a loan of a particular value with a specific repayment profile. The correct application of rules enables consistency and management by exception. For example, a skilled and experienced person need not be present at every consultation; he/she only needs to engage where their judgement and approval is essential.

Business rules are essential to the services sector: customers and their requirements can differ significantly (for example their health or financial situation), and rules ensure that decisions that are made are consistent with company policies and contribute to overall company success. In the latter years of the 20^{th} century, there was an increase in computer power that enables not only consistency, but monitoring, enforcement, and flexibility (Andreescu, 2008) and this process has continued (Schulte, 2018). These software systems, called business rules engines, enable organisations to build and maintain sophisticated sets of rules that can control and monitor many thousands of staff and millions of transactions, in real-time. They also allow rules to be changed to reflect changes in business circumstances. But while business rules deliver consistency, they do not automatically deliver efficiency or maximise customer service or revenue (Ross, 2016).

1.2 Business Rules and Business Processes

Business processes are important to both the services and manufacturing sectors. They are essential in manufacturing as this consists almost entirely of carrying out a series of tasks, in a specific order, in a repeatable way (Harmon, 2008). The human resources and physical resources which are used both cost money. The requirement to reduce costs has naturally led to the concept of business process optimisation where the sequencing of tasks and allocation of tasks to machines are planned to minimise costs or maximise revenue (Vergidis, 2008 & 2012). Indeed, manufacturing often goes a lot further and uses optimisation and forecasting techniques to maximise profit based on variable demand and anticipated demand in quite sophisticated ways (Tiwari et al., 2015). The main difference between manufacturing and services is that to be efficient manufacturing generally strives for repetition (i.e., to make large quantities of the same thing). However, there are manufacturing operations that deliver personalised products (either individually or mass customised). This ‘manufacturing a service’ or agile manufacturing segment bears some resemblance to the services sector in which the requirement is to manage different types of customer and deliver personalisation (Chaston, 2017). Effective management necessitates:

- correctness, that is, we implement the company policy and rules as designed and intended, and,
- consistency, that is, for each situation where a set of rules is to be applied, we use the same rules each time, every time.

Optimisation in the services sector is not so well advanced. Business rules are fundamentally designed for correctness and consistency, but not for optimised efficiency (Taylor, 2011). Two notable exceptions are staff rostering (Ernst et al., 2004) and supply chain optimisation (Hall et al., 2001). Rostering is widely used to have the right number of people (and no more than is necessary) with the right skills, in the right place at the right time. Cost reduction is motivation because people costs are dominant in the services sector. Supply chain optimisation is about activities, such as supply, storage and distribution, the allocation of resources to tasks, and task scheduling.

So, although an organisation may have the people required, with the right skills, all doing the same thing (in the same situation) using the same business rules, the challenge of doing better, with fewer people, at a lower cost remains. This challenge is the essence of the problem.

1.3 Context

Manufacturing productivity is generally defined as the value of outputs divided by the cost of inputs. It is further broken down into capital productivity (production per machine or factory) and labour productivity (hours of labour required to make a car). In services, measures such as added value, profit or revenue per employee are standard. Productivity improves with increases to the numerator (value of outputs), decreases to the denominator (cost of inputs), or both. Put another way; it is not just about reducing costs; increasing value has a similar impact. The latter is particularly important where the value of a transaction is variable.

Significant productivity improvements in manufacturing have been achieved through technologies such as automation, computer aided design and business process optimisation. There are a few vital objectives: mass production to reduce per unit costs of design and engineering; repeatability to increase quality and reduce rework; and continuous improvement to reduce the time taken, costs of labour and raw materials.

There are two sorts of services. Low values services, such as telecommunications and low-end catering and retail, resemble manufacturing insofar as scale and repeatability drive costs lower and increase productivity. High value or knowledge intensive services such as financial services and high end retail and hospitality are fundamentally different from manufacturing in two key respects; Firstly, customers are all different. Secondly, the service provided, the perceived value and the outcome of the service, depends on the information supplied by the customer.

Therefore, the question arises as to how to provide repeatability for low value services and responsiveness to customer information in high value services. One extraordinarily successful technology is business rules (Gartner, 2012). Business rules grew out of an earlier technology known as expert systems (Liao, 2005) where the idea was to capture or elicit expertise from an expert and create a rule base that would emulate the expert either automatically or in the hands of a non-expert. Expert systems developed into business rules which were more about consistency and repeatability. In principle, business rules do not supplant experts; they allow non-experts to deal with customers and issues and prompt them to seek guidance from experts when circumstances dictate.

Business rules are important to achieve good outcomes and thereby increase productivity. For example, in a sales situation information is gathered from the customer, and the salesperson processes this information correctly and reacts accordingly. The salesperson must understand what the customer wants (or needs), such as what they are prepared to pay, what flexibility they have, how sensitive they are to price, quality or delivery. For low value services, repeatability is essential as staff members are less qualified or experienced, and there is limited scope to personalise and a need to control the operation precisely. For example, in a fast food restaurant personalisation is limited to something simple like a special offer (such as upsizing) with a meal. In knowledge intensive service, much more information is elicited, and there is always the option, or necessity, to bring in an expert (such as an underwriter or sommelier) to deal with complex issues.

So, in these examples, business rules increase productivity and reduce costs by:

- empowering less qualified/less experienced and less expensive staff
- using more highly paid, expert staff, to improve service or increase revenue when the potential benefit justifies the cost

1.4 Motivation

Services account for most economic activity and employment in the developed world. For example, in Australia, services account for approximately 75% of GDP and 80% of employment (ABS, 2012).

The dominance of services represents a fundamental shift and creates a challenge for economies and organisations that have become accustomed to annual increases in productivity and standards of living that have, for the last 150 years or so, been driven by productivity increases in manufacturing. The new productivity challenge is how to increase productivity within services, and that of the knowledge workers that work within the sector (Drucker, 2018).

1.5 Research Strategy, Scope, and Outputs

1.5.1 Introduction

This thesis aims to contribute to the area of business rules research by defining and exploring a specific problem - that of Business Rule Optimisation – and creating a framework for its application to a broad class of problems. This chapter provides more detail on the aims, scope, objectives, and methodology employed. As an area of research, BRO is potentially as wide as BPO, which is the subject of over 400 research papers. It is therefore unreasonable to completely cover this area in a PhD thesis, so we:

- focus on a class of problems that includes many examples,
- consider types of rules that practical and recognisable to industry, and,
- use a case study methodology that delivers results that are meaningful and capable of supporting or catalysing further research.

This research is somewhat different form traditional research (Dawson, 2019) in two aspects:

i. It identifies a real business problem and seeks practical methods to find a solution.

ii. It is interdisciplinary, and the solution may include elements of machine learning, mathematical programming, probability, and draws on some research in psychology.

This thesis considers the broader question and then focus on a set of conventional business processes where the objective is, ultimately, to make a binary decision such as accept or reject, approve or not, or proceed, or not, for example. The concepts are then applied and proven on a representative case study involving credit approval. The research aims, objectives, questions and contributions presented should be read in this context.

1.5.2 Research Aim

This research aims to explore the concept of business rule optimisation and develop a framework for creating optimal business rules for a broad class of service business problems where the objective is to make decisions. The framework should be capable of representing the business process in a quantitative way and include essential factors that influence the quality of the decision and economic drivers. It should also include the human decision maker in the process and provide for their input when that is beneficial.

1.5.3 Research Objective

Business rule optimisation if a potentially broad area of research. The critical elements of this research are identified below:

1.5.3.1 Optimisation

i. To define what we mean by business rule optimisation and how that fits in with business optimisation and business process optimisation and other approaches that assist decision making in the process sector

ii. To define and understand the nature of the optimisation problem in terms of the objective function, degrees of freedom and constraints

iii. To identify practical methods of building optimal rules for the chosen class of problems

iv. To formulate the optimisation problem to include the factors that impact the objective function

1.5.3.2 Practical Implementation

i. To apply the theory to a case study example

ii. To develop a practical framework to create optimal rules for service business processes that result in decisions

1.5.3.3 Explore the impact of the research on applications

i. To identify the importance and potential of this approach by way of examples

ii. To identify a class of problems with similar characteristics for further analysis

iii. To test the framework on a different case to demonstrate effectiveness and applicability

1.5.4 Research Strategy

Robson (2002) defines two types of research strategies:

- Fixed design strategies that require specification before data collection; also known as quantitative strategies
- Flexible design strategies that evolve during data collection; generally referred to as qualitative strategies

Quantitative research is usually considered to be the traditional scientific research approach. It is an experimental model of enquiry, characterised by objectivity, reliability, and prediction. Much of the data collected and used is digital, and most of this research is done in a laboratory, where the environment and experimental conditions can be created and controlled. The key strengths of the quantitative approach are precision and control. Control is achieved through sampling and design; accuracy is achieved through quantitative and reliable measurement. The main limitation, concerning the real world, is that human beings are far more complex than a narrow view imposed by a quantitative approach (Burns, 2000).

Qualitative research is primarily based on an investigative approach, where much of the data collected is through interviews, surveys, and observation, and is in the form of words (Robson, 2002). Qualitative researchers tend to be involved in their study. As a result, the research questions and design tend to evolve, as more information is collected. Sociologists, psychologists, anthropologists, and more recently, business and industry, tend to use a qualitative research approach (Gummesson, 1991). The main strengths of the qualitative research approach are the insights gained from an inside view of the world under investigation and the researcher's involvement. These insights enable the researcher to derive unexpected observations from further examination. The main limitations and criticisms are validity and reliability. Data collection methods are time consuming, subjective, and prone to interpretation bias. The presence of the researcher causes bias during the collection of data. It is difficult to replicate studies; furthermore, it is difficult to make generalisations from the research findings.

This research is quantitative even when we include the human within the decision-making loop, and the optimisation process may be based on a forecast or inaccurate or incomplete historical data.

Besides, while defining and exploring a potentially large area, we are employing the Case Studies approach (Robson 2002) on an example of a class of problems with characteristics in common with many other types of service business processes.

1.5.5 Research Methodology

As identified above, this research is quantitative in and uses a pre-defined approach to analyse the available data. The main elements are:

i. Definition

The concept of business rule optimisation is introduced and put into context within the services sector.

ii. Literature Review

The objective of the review is to identify what has been done in business rules and related research, such as business process optimisation and re-engineering.

iii. Application

Some example service business problems are explored to assess the applicability and relevance of the approach and to determine the existence of a genuine optimisation problem.

iv. Categorisation

We identify a wide range of problems with common characteristics for further, more detailed analysis, including problem formulation and consideration of all the relevant aspects that impact the objective function.

v. Methodology

We identify **practical** methods, tools and techniques that can be used to address the business rule optimisation problem. By practical, we mean proven, accessible methods that can be used to create business rules in a format recognisable to the business operator or manager.

vi. Case Study

From the chosen category, we take a case study that has the relevant characteristics of our category and, using real data, prove the concept and identify potential benefits.

vii. Framework

Using the insights from the case study, we develop a **practical** framework for optimisation of similar problems.

viii. Validation

The final stage is to take a different example and apply the framework to confirm its effectiveness

1.5.6 Research Scope

A business rule is a statement that defines or constrains some aspect of the business. It is intended to control or influence the behaviour of the business, and the objective of a business process is to produce the desired result. In other words, business processes deliver results for the business, and business rules influence the business processes. Therefore, we can conclude that *business rules influence business results*. We can further conclude that, unless all business rules give the same results, there will be a set (not necessarily unique) of business rules that will deliver the best results. Finding that set of rules is an optimisation problem, and this is a different problem than the traditional optimisation approach where a specific situation is optimised. For example, in manufacturing, we can optimise production based on the orders received in a day, week or month (Tiwari et al., 2015). Furthermore, if we look at the definition of optimisation, we see two distinct meanings (Dictionaries, C. 2009)

1. To optimise a *situation or opportunity* means to get as much advantage or benefit from it as you can. For example, see Mousavi et al. (2019)
2. To optimise a *plan, system, or machine* means to arrange or design it so that it operates as smoothly and efficiently as possible. For example, see Osman et al. (2018)

Another way to look at this is:

1. Data driven, automatic or on-line optimisation where the optimal actions are determined based on the current situation
2. Static or off-line optimisation where the best action for each foreseeable situation is determined in advance

In this research, the main objective is to consider the second type of optimisation, that is, to optimise the system (that is the business rules) so that the business processes operate in such a way that the best results are obtained over all anticipated situations, not just in one. In this context, of an ongoing process, the results are never complete, so we can only target the best *expected* results. Therefore, we optimise *expected values* using some sort of forecast, or historical data, or a combination of both. However, we also recognise that there are requirements for situational optimisation should the actual differ from the expected, and we need to have a way to modify the rules. This adjustment can be on-line when we have more, or, less, or different customers to that anticipated and applying the same rules may lead to long queues or under-utilisation of resources. Or for the long term when we determine that a change in customer requirements or characteristics means that we need to revise the rules. The latter is effectively a repeat of the off-line optimisation using more recent (or additional) data on customer transactions.

A potential downside of an optimised rule set is one of specialisation: the organisation becomes very adept at (and profitable from) servicing the customers it had or expects. Without adjustment, a change in customers could have a much more marked impact than if the rules had never been optimised. Failure to recognise this could potentially lead to failure of an organisation.

As we shall see in Chapter 2, business rules include those around constraints and policies, that are non-negotiable, and computation, reasoning, and allocation (CRA) rules. CRA rules may be changed to improve business performance in terms of profit, revenue, or customer service.

1.5.7 Case Study

There are many examples of CRA business rules, and we have identified an example problem – Loan Approvals - to test the ideas, develop a process, and illustrate its application:

i. Where decisions can be made by the rules, and experts, or both

ii. Would typically (or could) use CRA rules

iii. Where we can obtain data on judgements and outcomes

Experts typically make loan approval decisions, so there is scope for at least some to be made by rules. The approval process typically uses computation rules (for example calculating loan/valuation or loan/salary), reasoning rules (determining acceptance), and allocation rules where the path of the customer through the process is determined by the information provided. We also need to calculate how many experts are required based on the expected caseload and case-mix. Finally, we need a way to modify the rules so that we make the best use of the available experts when the caseload or case-mix varies from that the expected.

There is a large, publicly available data set (Kaggle, 2020) that includes data on judgements and outcomes. This data will enable us to model the current decision-making process (the expert), develop rules based on outcomes, compare the accuracy of both, and then decide which is the best option on a case-by-case basis.

1.6 Research Questions

Business rules can be applied to a wide range of businesses, and the questions and hypotheses below are relevant to all business processes.

However, in this thesis, we limit our scope as follows:

iv. We consider a wide range of service business processes with common characteristics – but not every service business process - and then identify a representative case study to prove the concept

v. We want to find practical methods of implementation that use reliable, widely accessible tools

vi. The rules we build should be of a form that is recognisable by industry

vii. The rules should be such that their conclusions are traceable and capable of being understood

The questions, hypotheses, and contributions, below, should be read in this context.

We have used data on judgement and outcomes, including the potential for a human decision maker to work with the rules, and the impact of gathering information on the customer.

In this context, the overarching research question is:

How can business rules and be designed for best business results?

The underlying hypothesis is that there are real-world examples where this can be done.

To further understand this overarching question, several subsidiary questions are generated:

RQ1 How is the optimisation problem to be defined?

> We identify a model of an example of a set of service business processes, an objective function, constraints, and degrees of freedom.

RQ2 How should the rules be built and optimised?

> We investigate the processes and procedures that are capable of creating and optimising business rules and determine the best options for our example.

RQ3 What information should be requested initially and subsequently?

> We determine the cost of obtaining and processing the information and balance these against the potential benefits of having more information upon which to base decisions.

RQ4 How should business rules be built using data on expert decisions and outcomes?

In our case study application, data on decisions is more readily available; outcome data may take years to collect and is much rarer (at least in the public domain) and is only available for cases that have been accepted. However, when we have data on outcomes, we need a way to use it to best effect.

RQ5 How should we incorporate a human decision maker (expert) to best effect?

Business rules are designed to reach a conclusion or decision that will be implemented. How do we build, and optimise, rules that have the additional option to refer to an expert that will reach a conclusion or decision?

RQ6 On a case by case basis, how should we decide when to refer the decision to a human?

Each case may have different attributes that are used by the rules to decide. But they may also have other characteristics that impact the economic optimum.

RQ7 How can the rules be adapted to maintain maximum efficiency in changing circumstances?

We propose to create a set of business rules that will be optimal for an anticipated range of scenarios. Still, in practice, there will be short- and medium-term differences between actual and expected. We find a method that enables the rules to adapt to a new situation and take advantage of that opportunity to do better (than they otherwise would have done).

These research questions imply the existence of a set of optimal business rules that support the business process. Besides, the last question suggests that some operational adjustment is possible to maintain optimality as circumstances change.

H1 The underlying hypothesis is that it is possible to optimise business rules in the sense that they give the best results over a defined range of situations (either determined by analysis of historical data or forecasts) considering:

- Data on outcomes and judgements
- The potential to refer decisions to experts

- The costs of asking for further information
- The impact of further information

H2 A secondary hypothesis is that business rules can be adapted in an operational sense to take account of circumstances that differ from those already experienced or anticipated.

1.7 Research Contribution

The contribution of this work is an analysis of the factors that should drive the design of business rules that maximise productivity, efficiency, and profitability of an organisation. Previous research is limited to single disciplines (for example, machine learning, psychology), and can be characterised as bottom up (extension and addition), and necessarily fragmented. This thesis:

RC1. Introduces the concept of business rule optimisation problem and identifies ways to solve the problem examples from the services sector.

RC2. Creates an extended version of the LENS model (Brunswik, 1985) where we have outcomes, the expert decision and, now, additionally, the rules decision.

RC3. Creates a model of transaction abandonment as a function of time using the Weibull distribution (Evans et al., 2000), and incorporates this model into the feature selection (Hall et al., 2013) and rule building problems.

RC4. Applies business rule optimisation to a representative example in the services sector problem (that includes potential profits, losses, costs, and transaction abandonment) demonstrates that it can be feasible and useful.

RC5. Creates a framework and guide for building rules for the general problem of customer selection and acceptance that is applicable for credit approval and similar classification problems.

Some of this work has been published:

Dormer, A. (2012). Optimising business rules in the services sector. Int. J. Soc. Behav. Educ. Econ. Bus. Ind. Eng, 6(10), 2580-2584.

> This paper introduces the concepts and ideas and uses examples from the services sector (credit approval, debt recovery and transport) to define the

optimisation problem in each, and show that there is a mathematical optimum, even in simple examples.

Dormer, A. (2017). A Framework for Optimising Business Rules. In International Conference on Business Information Systems (pp. 5-17). Springer.

This paper applies machine learning to the initial assessment stage of the Lending Club problem, defines the optimisation problem (RC1), and creates a method to combine business rules and human experts, and demonstrates potential benefits (RC4).

Dormer, A. (2018). Cyborganisation: Machines and Humans Make Optimal Decisions Together. In Third International Congress on Information and Communication Technology: ICICT 2018, London (Vol. 797, pp. 487-497). Springer.

In this, we introduce the extended LENS model (RC2) and use it to combine rules and experts. We use data from Lending Club to demonstrate that the approach is feasible and potentially beneficial.

Dormer, A. (2018). Business Rule Optimisation: Problem Definition, Proof-of-Concept and Application Areas. In International Conference on Business Information Systems (pp. 51-62). Springer.

This paper combines the results of the previous papers to define the more general problem and identifies other areas of application. We also recognise methods to deal with transaction abandonment, feature selection and rule adaptation (RC5).

1.8 Research Outputs

The critical research output is a framework for optimising business rules that support a business process that incorporates several important features. These are common to a class of service business problems that result in a decision, ordinarily binary, such as approval (or not), acceptance (or not), act (or not), etc. The framework supports several key elements:

- Creating a model of the human decision maker
- Creating a set of rules that can be understood and implemented in a practical way

- Modelling the cost of processing information as a function of the amount requested
- Modelling the tendency for transactions to be abandoned as more information is requested
- Optimising the interaction between the rules and human decision makers
- Optimising the operation of the rules when caseloads or the case-mix vary from the anticipated or historical situation

1.9 Summary

This chapter sets out the research aims and strategy appropriate to the BPO problem. BPO is a potentially large field of quantitative research, and we employ a methodology that defines the broader problem, with examples. We then identify a category of problem that is broad yet representative upon which to develop the theory. The case study approach (Robson, 2002) proves the concepts and enables the development of a framework.

1.10 Thesis Overview

This thesis consists of five parts:

i. Chapters 2 introduces the problem of business rule optimisation explores previous research concerning business rules and other relevant areas. Chapter 3 explores some simple examples of business rule optimisation in loan approval, debt recovery and transport. These are selected to highlight the trade-offs and potential for optimisation, even in the simplest of situations.
ii. Chapters 4-7 set out the theoretical basis and contribution of this thesis in the context of CRA rules applied to common business processes and the case study example. They include rule scope, customer categorisation, through interaction, adaptive information gathering and minimising customer abandonment, and the constructs used in the business rules themselves.
iii. Chapters 8-10 include a description of the case study and uses real data to assess the proposed approach. They consider the impact of human judgement and abandonment and classify rejected customers. The last chapter presents an adaptation strategy to account for differences in caseload and case mix.

iv. Chapter 11 sets out a practical framework and step-by-step process to apply the findings of this research to a real-world business problem.

v. Chapter 12 consists of conclusions, limitations of this research, and areas for future work.

2 LITERATURE REVIEW

2.1 Introduction

This literature review explores previous work related to business rule optimisation. By business rule optimisation we mean the process of building business rules that will maximise or minimise the expected value of a quantity (such as profit cost) when they are applied to the customer orders or enquiries that we forecast or expect to receive.

We first explore the various definitions of business rules to understand the key concepts and how business rules are related to business processes and how they influence business outcomes. Then we review research in business rules, business process optimisation, and other relevant areas such as human decision making to determine what has been done. Finally, we identify the research gaps and potential areas of research.

2.2 Business Rules

2.2.1 Definitions

There are many definitions of business rules, generally from a business or information technology (IT) perspective, the latter deriving from the requirement to program rules as part of enterprise systems.

- Business rules are declarations of policy or conditions that must be satisfied (Martin et al., 1998)
- The core notion of a business rule, common to all sources, is that it is a constraint on the behaviour of an enterprise; it specifies what is allowable and what isn’t. (Liete, 1998)
- A business rule is a statement that defines or constrains some aspect of the business. It is intended to assert business structure or to control or influence the behaviour of the business. (Hay et al., 2000)

- Business rules represent projections of organisations' constraints and ways of working on their supporting information systems (Kardasis, 2004)
- A business rule is a rule that can be interpreted by computers, that defines or restricts some aspects of a business, introducing obligations or needs, according to organisational policies and rules (Kamada et al., 2007)
- A “business rule” is a directive of a domain which controls the conduct of a business activity of that domain. Its goal is to structure a business activity (policy, know-how) to control or influence the conduct of a business activity of the domain in question, in view of achieving an expected result (Roger et al., 2010)

The common theme in all these definitions are notions of obligation and constraints, which is not surprising given that rules in real life are about what we *should do* and what we *should not do*. In an optimisation context we can relate these to feasible (allowable or valid) and infeasible (not allowable or invalid) solutions (where the term constraint, unfortunately, has a different meaning). For example, we may be *obliged* to provide telecommunication service to customers from the whole country, but we are *not permitted* to offer a discount for city dwellers. To a lesser extent, there is the notion of expected results and needs, that is, what outcome(s) or effect (s) do we want to achieve, that is, business rules are a means to an end. So clearly rules have a direct impact on the operation, results and outcomes of a business.

For example:

- External rules and regulations, such as minimum wages or opening hours for a business, have direct impacts on costs and revenue
- Internal rules, such as acceptance or rejection of customer orders, how to process complaints, etc., impact revenue and risks

In this thesis, we will use the definition that rules are a framework within which a business operates, but they do not define what operations are performed.

There are standards for the expression of business rules such as SVBR, (Object Management Group, 2011) or Rulespeak, (Ross, 2006) which can be used to express examples consistently. According to Packt (2009), rules can be categorised as follows:

Business Policies: These are rules associated with general business policies of a company, for example, loan approval policies, escalation policies, and so on.

Constraints: These are the rules which business must include, and work within the scope of while going about their operations. Rules associated with regulatory requirements will fall under this category. Another term for this is compliance; rules must enforce compliance and avoid non-compliance; compliance is a given.

Computation: These are the rules associated with decisions involving any calculations, for example, discounting rules, premium adjustments, and so on.

Reasoning capabilities: These are the rules that apply logic and inference course of actions based on multiple criteria. For example, there may be rules governing the up-sell or cross-sell of products based on the customer profile.

Allocation Rules: Some rules are applicable in terms of determining the course of action for the process, based on information from the previous tasks. They also include rules that manage the receiving, assignment, routing, and tracking of work.

The problem that we are dealing with in this research includes rules from these categories. Business policies and constraints (or more generally compliance) may be considered first as we assume compliance to be a given. For example, in loan approval, there are regulatory and policy considerations around the age of a borrower. From a compliance perspective, we disqualify a potential customer who is too young or too old, for example, before we get to the point of assessment.

Business rules deal with variability and uncertainty. Customers are different; outcomes are often uncertain. Yet we need consistency in our business processes. For example, if two customers have the same attributes, it is not acceptable (or indeed sensible) to treat them differently. As identified in Section 1.1, there has been an increase in computerisation that enables not only consistency but monitoring, enforcement, and flexibility. These software systems, called business rules engines, allow organisations to build and maintain sophisticated sets of rules that can control and monitor many thousands of staff and millions of transactions, in real-time. They also enable rules to be changed to reflect changes in business circumstances. But while business rules deliver consistency, they do not automatically deliver efficiency or maximise customer service or revenue (Ross, 2016).

2.3 Business Rules Ontology

To understand the anatomy of a business rule, we can divide a business rule primarily into the following four blocks (Kay et al., 2000):

Definitions of Terms: This helps in providing a vocabulary for expressing the rules. Defining a term acts as the category for the rules. For example, customer, car, claims, and so on identify the entities for the business.

Facts: These are used to relate terms in definitions with each other. For example, a customer may apply for a claim.

Constraints: These are the constraints, limitations, or controls on how an organisation wants to use and update the data. For example, for opening an account, a customer's passport details, or social security details are required.

Inference: This applies to logical assertions such as 'if X, then Y' to a fact, and infers new facts. For example, we have a single account validation rule (if an applicant is a defaulter, then the applicant is high-risk). And we know that Harry (the applicant) has defaulted earlier on his payments for other bank services, we can infer that Harry is a high-risk customer.

Based on the above definitions and categorisations, there are three essential characteristics that we should be concerned about with rules:

i. Rules can define constraints, such as statutory constraints (e.g., maximum working hours) or business policies (e.g., acceptance criteria for new customers).

ii. They are controlling the flow of the business process between tasks and resources. For example, IF (condition) do Y or go to A, ELSE do X or go to B. These are examples of CRA rules.

iii. They are enforcing actions such as the it_is_necessary_that constraint. For example, to make progress a claim it_is_necessary_that the claim form has been received; essentially preconditions for a response to become applicable.

We want to optimise the rules, but we do not want to change rules that should not be altered or cannot be changed. So, there appears little point in adding in rules of type iii into our rules set, as we cannot make any changes to them. We can assume that they have already been applied.

2.4 Business Rule Optimisation

An organisation may have the right number of people, with the right skills, all doing the same thing (in the same situation) by using the same business rules. This study is seeking ways to change the business rules so that the organisation using such rules could be more profitable, serve more customers, or serve its customers better.

Business rule optimisation is a relatively new area of research and the definition used in this thesis is: Business rules optimisation (BRO) is about **finding** that set of business rules that maximises the **expected** net contribution to the organisation that uses them.

Hence, we are concerned with the **structure** and the **parameters** within the business rules as far as they impact business performance. We are not interested, for example, in:

- The efficient construction of business rules from expert knowledge (Sneed, 1996), (Gottesdeiner, 1999) or other data sources (Shao, 1996),
- The creation (Chikofsky, 1990), (Chisholm, 2004), organisation (Kardasis, 2004), deployment (Rosca, 2002), or integration of business rules (Cibrán, 2003).
- How they enforce policies (Leonardi, 1998)

Taylor (2011) covers rules and optimisation, but in the context of rules working with optimisation (with techniques such linear programming) to provide decision support; each case or situation is dealt with by rules and optimisation with a separate optimisation calculation carried out each time. We are looking at the ability to optimise the rules (in advance) so that they get the best expected outcome, without further optimisation being required. This process is an example of off-line or static optimisation. While this approach will not provide an optimal outcome for each situation, it removes complexity from the operation where business rules are more straightforward, quicker, and generally more reliable than an optimisation calculation for each new situation.

We define situational (or online) optimisation as choosing that set of inputs (usually real, binary or integer variables) that optimise an objective (or objectives) such as profit, productivity or revenue for a specific set of circumstances (such as customer orders, available staff and equipment, etc.). There are many methods to solve the situational optimisation problem, including gradient methods, constraint programming and genetic algorithms, for example. Rule optimisation is different as we are optimising over the domain of all potential business rules that could be applied. The general problem is one of optimising a functional (a function of a function) with the first function being the objective (which is fixed), and the second function being the rules (with an infinite range of candidates).

If we limit the rules to a given structure, then we can reduce this problem to one of identifying which rules to apply (binary or integer) and their parameters (real). In this way, we can solve the rule optimisation problem with the same proven techniques that address the situational problem. This problem is explored further in Chapter 5 with references for, and examples of, static/offline optimisation (5.2.1) and situational/online optimisation (5.2.2).

Research on BRO is limited. A search revealed two US patent applications which relate to the interactions between participants in a communication channel (such as telephone calls or social media) (Gupta, 2011, Gupta, 2013). Optimising business rules for fraud prevention is described in (Liu, 2014). Optimising rules for configuration studied in (Jandir, 2009), and (Begunov, 2008) is about the simulation of social-economic systems within a city. None of these studies addresses or considers the concept of maximising expected value, which is fundamental in our research. Besides, Čubrilo et al. (2016) and Wang et al. (2014) identify gaps and shortcomings concerning business rules and the integration of business rules and business processes.

To relate our work to previous research in business rules, that is not explicitly referred to as optimisation; we categorise as follows:

- Recognition that business rules have an impact on organisational performance
- The ability to modify rules to change performance
- Dealing with uncertainty and variability
- The combination of business rules and human experts

2.4.1 Impact of Business Rules

There is research that recognises that business rules have an impact on an organisation. These studies focus on issues such as flexibility (Van Eijndhoven, 2008), consistency, (Gottesdiener, 1997), the effect of change and change management (Bajec et al., 2005) and compliance (Kruk et al., 2003). Antonius et al. (2014) consider how business rules should be managed and maintained to minimise resource utilisation and costs. Wei et al. (2017) examine the relationship between business rules and business processes from the perspective of whether it is better to integrate or to keep them separated. Wang (2017) proposes a way to integrate business rules and business process models to create a complete representation for business process modelling. He refers to optimisation, but of the business process, not the rules, and the rules are considered as constraints rather than potential opportunities. None of these studies addresses the broader issues of the impact on a business and how to achieve the best expected outcome.

2.4.2 Changing Business Outcomes by Changing Rules

To date, several studies have investigated the means whereby rules can be changed to achieve specific outcomes or Key Performance Indicators (KPI’s). Kunz et al. (2015) address the optimality of retail pricing and the impact of the constraints imposed by business rules. Different business rule sets have been analysed to improve the balance between inventory and operational effectiveness in supply chain logistics (Oswald, 2013). Rules can also be used in a matching algorithm to allocate work to a range of human decision makers based on their skill levels and availabilities (Quinzaños et al., 2014). In a similar vein, Yan et al. (2015) examine the use of business rules for service selection.

All the above are examples of business rule manipulation to achieve a single objective such as pricing, inventory or resource allocation. They do not have an overall objective, nor do they optimise expected values.

2.4.3 Uncertainty, Variability and Unpredictability

Cuzzocrea et al. (2014) consider the issues of uncertainty and scalability in business rule applications. Hegazi (2015) utilises fuzzy logic in business rules to deal with uncertainty and ambiguity. In BRO, as we have defined it, there is variability and unpredictability, as we are optimising expected values using a set of historical or forecast data. Also, in the case study and similar business processes, there is a degree of uncertainty as decisions are never always correct.

2.4.4 Integration with Experts

Ninan et al. (2014) apply planning and business rules to the health sector for optimal caseload allocation based on requirements and skills. Wang et al. (2016) consider the problem of how business rules can be modelled and manipulated to meet a single KPI; in this case, the percentage of banking customers accepted. This paper is like our case study, as in that research, there is a consideration of what cases should be referred based on an assumed distribution of customers. Still, there is no consideration of the accuracy of the rules or the experts. By integration with experts, we mean integration at the execution stage, that is when the rules are working. There is another form of integration called Human-in-the-Loop AI/Machine Learning (Xin et al., 2018); this is fundamentally different and concerned with integration at the learning or validation stage. There are also applications where the human is part of the decision process, for example overseeing an algorithm and helping it learn (Subramania et al., 2011), providing data to an algorithm (Pinto et al., 2013) or being part of every decision (Cao et al., 2010). Amgoud (2009) presents a qualitative approach to decision making with both upside and downside that is relevant, but for rule optimisation, we also need a quantitative method. For the selected rule optimisation problem, we require to understand the relative merits (in terms of financial or customer outcomes) of the business rules or human expert making the final decision.

2.4.5 Conclusion

In summary, however, none of the previous business rules research:

- Model the accuracy of the rule decision or the expert decision
- Investigate the combination of automated and human decision-making in the context where the resources deployed must be balanced against their impact on the outcome
- Optimise the rules against their overall impact on the bottom line, combining all three business rule optimisations summarised above

Čubrilo et al. (2016) make the case that business rules are not fulfilling their true potential. According to Čubrilo:

> In the context of business processes and rules, their modelling and effective application to business practices, there is a paradox of the existence of highly developed theories which would enable quality

> practices, and a whole array of individual, very proficient practical solutions (implementations of business rules) on the one side, but no universal and generally accepted methodology for modelling business rules as a service for business processes and information systems to support them on the other side.

Also, Wang et al. (2014), identifies similar problems.

> While we sense an increasing argument in literature for business process and business rule integration, we observe a gap in the body of knowledge whereby the benefits, current approaches and maturity of existing research have not been consolidated and is thus not well understood

2.5 Business Processes

Business rules set the framework, but things need to be done to achieve the objectives and desired outcomes. This collection of tasks is often referred to as business processes.

In common with business rules, there are many definitions of business processes.

For example:

- A business process is a set of linked activities that takes an input, and it transforms it to create an output. It should add value to the input and generate an output that is more useful and effective to the recipient (Johanson et al.,1993)
- A *business process* is defined as the chain of activities whose final aim is the production of a specific output for a customer or market (Davenport 1993).
- A *business process* is a set of one or more linked procedures or activities that collectively realise a business objective or policy goal, normally within the context of an organisational structure defining functional roles and relationships. (Fan, 2001), and (Shen et al., 2004)
- The term *business process* is used to denote a set of activities that collectively achieve a certain business goal. Examples of these processes are the hiring of a new employee or the processing of an order (Castellanos et al., 2004).
- A business process is the combination of a set of activities within an enterprise with a structure describing their logical order and dependence whose objective is to produce the desired result. (Aguilar-Saven, 2004)

- A business process is a set of partially ordered activities, which produce a specific product or service that adds value for a customer. (Ratkowski, 2012)

All of these refer to tasks or activities towards a goal or desired outcome. There are also references to linkages (between activities) and ordering or sequencing.

In this thesis, we can view business processes living within business rules (the constraints and obligations) towards the outcomes and results we require, and we can also have business rules within tasks and activities. But this is not a simple relationship as rules may apply to tasks, activities (collections of tasks) or business functions.

2.6 Business Process Re-Engineering and Optimisation

2.6.1 Business Process Re-engineering

These techniques are generally all about ways to cut costs, improve quality, improve throughput, etc. For example, Six Sigma is used to reduce variability and improve quality, (Weiner et al., 2004) and the Theory of Constraints (Tulasi, 2012) is used to increase throughput by removing bottlenecks.

Business Process Reengineering (BPR) involves the radical redesign of core business processes to achieve improvements in productivity, cycle times and quality. In BPR, companies start with a blank sheet of paper and rethink existing processes to deliver more value to the customer. They typically adopt a new value system that places increased emphasis on customer needs. Companies reduce organisational layers and eliminate unproductive activities in two key areas. First, they redesign functional organisations into cross-functional teams. Second, they use technology to improve data dissemination and decision making. BPR can be regarded as a precursor to the more rigorous approach of business process optimisation (Vergidis, 2008).

2.6.2 Optimisation

Mathematical optimisation consists of finding the maximum (or minimum) of a function (the objective function). The objective is a function of one or more variables (degrees of freedom) that are typically real, binary or integer. Some constraints must be respected, and the constraints also depend on the degrees of freedom. Methods to solve an optimisation vary depending on the nature of the objective function, constraints, and the variables (Yang, 2008). If we confine our attention to real variables we have:

Linear programming is the oldest method and is used when the objective and constraints are linear functions (Dantzig, 1998)

Quadratic programming is also very reliable and requires the objective function to be quadratic, with the constraints generally remaining linear (Wolfe, 1959)

Non-linear programming covers all other problems, and the difficulty and applicable solution methods depend crucially on the nature of the objective function and the constraints. Key issues include convexity, continuity, and differentiability (Dorn, 1963).

Binary or integer variables (so-called combinatorial problems) are much more challenging to solve and may never solve, even for small linear problems. Different approaches are taken. With classical mathematical programming, the idea of relaxation is used where the binary and integer variables assume real values, and then systematically fixed until an integer, or binary solution is obtained. New solutions are then tried that either result in further integer solutions or rejection by comparison with the best solution so far (Beale, 1979). Direct search methods – that only use the value of the objective function – explore solutions using just binary or integer variables (Wright, 1996). Many of these employ ideas from the natural world such as simulated annealing, ant colony or evolutionary computing.

As identified in 2.4, the general business rule optimisation problem is optimising a functional with infinite dimensions. Even if we can identify all the potential rules we still have a combinatorial problem (the existence of a rule) with real variables (parameters employed in the rules) a non-linear objective (made up of economic, statistical and/or empirical elements).

2.6.3 Business Process Optimisation

Business process optimisation has grown up as a branch of industrial engineering which has been extensively applied to production processes and concerned with issues such as quality and efficiency. Provided that quality is maintained, the key issues are time and cost, which are functions of resource, labour, and equipment utilisation. Efficiency and quality in a production process are critically dependent on repeatability.

Business Process Optimisation (BPO) is the problem of finding or constructing that set of business processes that are:

- feasible concerning any constraints imposed (in the broader sense of what we have to do, and what we should not do), and
- maximise or minimise the desired objective(s) such as profit, revenue or time. Laguna et al. (2005) explore modelling, simulation and design of business processes, with a simple example of optimisation using evolutionary computing.

There is a large body of research on BPO. BPO research focusses predominately on the processes required to produce an outcome at minimum time or cost, and the way that tasks and activities are structured (such as ordering, linkage). In almost all cases in the literature, rules (where cases or components are directed one way or the other) are not considered.

Vergidis (2008) provides a literature survey of business process optimisation and applies multi-criteria, evolutionary optimisation to the general problem of selecting, ordering, and linking business processes to create an optimal set.

His conclusions at that time were:

- The current trend in business process modelling is the use of diagrammatic models that visualise the business process but do not provide the necessary quantitative constructs for performance analysis and optimisation.
- The proposed classification demonstrated a lack of support by most business process modelling techniques for structured process improvement
- The few business process optimisation approaches reported in the literature are highly complicated and yet address only simple sequential business processes.
- The author notes that business processes are more than just scheduling and include business rules. Still, business rules per se are not considered in either the modelling or the optimisation problem.

More recent research (Vysockis, 2018), for example, still refers to this framework and focuses on solving more complicated, stochastic, problems. Business rules are considered, but while the rules themselves are part of the model, they are not considered the subject of optimisation.

Vergidis (2008), examines business processes modelling methods such as diagrammatic representations and formal mathematical models, and their suitability as a basis for optimisation, and uses evolutionary computing and a multi-criteria objective function. But the problem is still simplified, and research focuses on representations and business processes represented as nodes without consideration of changing the contents of the nodes, for example, the rules. Stelling (2006) and (2009) describes a method of expressing business processes as chromosomes and then applies an evolutionary computation-based technique to optimise the selection and connections between functions, activities, and tasks. The essence of evolutionary computing (or genetic algorithms) is to create a process that emulates evolution. Initial candidate solutions are produced and then a process of combination and mutation is applied, with the best solutions retailed and remainder discarded. To do this effectively, we require a coding scheme (like the genome) that supports combination and mutation. In BPO applications the procedure may consist of creating a taxonomy of activities, tasks, links, and resources, and then assigning integer codes to each to create a representation of any process. This methodology has the potential to optimise business rules. To do that we represent the rules as processes with outcomes dependent on their inputs. But the BPO problems solved are quite limited, mostly looking at substituting or adding a couple of tasks. Table 1 and Table 2 show a classification scheme for business processes from domains (the highest level) down to sub-transactions (the lowest level). As identified above, this enables the application of evolutionary computing to solve the optimisation problem.

CODE	LEVEL	NAME
1	DOMAIN L1	HUMAN RESOURCES
2	DOMAIN L1	CUSTOMER SERVICE
3	DOMAIN L1	FINANCE
4	DOMAIN L1	INFORMATION TECHNOLOGY
1	DOMAIN L2	PROFESSIONAL SERVICES
2	DOMAIN L2	ASSURANCE
3	DOMAIN L2	CONTACT
4	DOMAIN L2	FULFILMENT
5	DOMAIN L2	BILLING
6	DOMAIN L2	GENERAL
01	DOMAIN L3	SERVICE ESTABLISHMENT
02	DOMAIN L3	QUOTES
03	DOMAIN L3	ORDERS
04	DOMAIN L3	FAULTS
05	DOMAIN L3	FIND PRODUCT
06	DOMAIN L3	COMPLAINTS
01	ACTIVITY	PLACE ORDER (TELEPHONE)
02	ACTIVITY	PLACE ORDER (ONLINE)
03	ACTIVITY	TRACK ORDER
04	ACTIVITY	MODIFY ORDER
05	ACTIVITY	CREDIT APPLICATION (LOW)
06	ACTIVITY	CREDIT APPLICATION (HIGH)
01	TRANSACTION	RETRIEVE CUSTOMER DETAILS
02	TRANSACTION	RETRIEVE ORDER DETAILS
03	TRANSACTION	EMAIL ORDER STATUS

Table 1 Taxonomy Coding System (Steen, 2010)

CODE NAME	NO. DIGITS	EXAMPLE/EXPLANATION
Domain Level 1	1	1=Human Resources
Domain Level 2	1	5=Billing
Domain Level 3	2	04=Faults
Activity	2	03=Track Order
Transaction	2	03=Order Product
Sub-Transaction	2	03=Check Postal Code
Input 1	5	Up to 3 possible alternative inputs of 5 digits E.G.002050020600000=Placed Order or Placed Delivery
Output	15	Up to 3 possible alternative outputs of 5 digits E.G. 002030020400000=Pending Order or Pending Delivery
Constraint	9	Up to 3 constraints of 3 digits E.G. 001003000=Must be first step AND must be linked to next step
Resource	3	001=Internal Customer RDB
Flows In	10	Up to 5 incoming flows of 2 digits per flow (normally 1 except for join (AND-Join) & merging (OR-Join) process steps). E.G. 0101000000=2 incoming normal flows
Flows Out except for fork (AND-Split) & decision (OR-Split) process steps) E.G. 0109010000=1 outgoing normal flow, 1 sequence flow loop, 1 outgoing normal flow	10	Up to 5 outgoing flows of 2 digits per flow (normally 1
Previous Step	50	Allows for up to 5 steps of 10 digits (Domain L1, Domain L2, Domain L3, Activity, Transaction, Sub-transaction)
Next Step	50	Allows for up to 5 steps of 10 digits (Domain L1, Domain
L2, Domain L3, Activity, Transaction		

Table 2 Coding Details (Steen, 2010)

Steen (2010) describes a method of generating optimal business processes from a set of business rules. He describes a four-step approach of rules creation (from the specification of the rules), process creation, optimising the process and then creating the final business process model. But the rules are an input, not subject to any optimisation process. And the optimisation of the processes is around resource allocation, performance, and costs, which is the typical objective in BPO.

Vergidis (2012) presents an optimisation framework for generating optimised business processes with diverse designs by ordering and linking tasks allowing parallel activities and branching. But again, there is no reference to business rules.

Kamrani et al. (2012) explores business process optimisation by considering the assignment of agents to tasks.

A missing theme in BPO research is consideration of differences between customers or externalities and the impact that these have on the process and the objective function. There is no reference to the probability distributions of the 'inputs'. The objective(s) are not expected values or values associated with alternative scenarios, or indeed probability distributions like the inputs, which would be the case if we were optimising a set of business processes that specifically deal with variations. Even though examples are drawn from the services sector, these are mechanical processes such as order processing. Gibillini (2008) identifies some of the shortcomings of BPO as applied to the underwriting function of the insurance sector, which is an excellent example of optimising a process for more than just speed or cost. She identifies form validation as one of the top five obstacles:

> These key decision points carry liability when either the timeline is missed or an opportunity to validate need is lost. Further, time constraints foster "approve-or-deny" practices, affecting the quality of case management and restricting the time available to engage in validation activity.

Aghdasi (2010) explores the impact of business rules on business process optimisation and explores the concept and a simple example in the dairy industry. It makes the point that most research in BPO considers only time and cost and does not consider business goals which can be expressed through business rules. It also identifies that work is required to explore the ordering and creation of business rules. In the example, a rules-based approach is used to optimise parameters of rules, and there are outcomes including acceptance, sampling* and acceptance.

*In this case 'sampling' is the same as referral, that is we cannot decide at this point, and something else must happen before a decision is made around acceptance or rejection.

The optimisation criteria adopted are expressed as alignment with business goals. We (2012), give examples within the services sector, including transport and financial services where optimisation of the rules is to maximise expected overall profits. The author makes the point that business rules optimisation must consider *expected values*. Even Aghdasi (2010), which is concerned with rules and optimisation, the concept of expected values is absent.

These discuss how (1) rules influence business outcomes, (2) we can change (and even optimise) business outcomes by either modify existing rules or creating new ones. But in both cases, the examples are deliberately simplified.

2.7 Knowledge Engineering & Expert Systems

Business rules evolved from early attempts to emulate experts in so-called expert systems (Gottesdiener, 1997). The requirement to elicit, codify and manage the necessary expert knowledge led to the field of knowledge engineering (Schreiber et al., 2000). An expert system consists of a knowledge base and a rule engine, like present day business rules. Still, the challenge with genuine experts was the extent of their knowledge and the difficulty of explaining what knowledge was used and how individual decisions were made. Not knowing what knowledge or reasoning was used to decide makes it hard to build the knowledge base or rule engine, respectively.

TechTarget (2017) gives a good summary:

> Knowledge engineering is a field of artificial intelligence (AI) that tries to emulate the judgment and behaviour of a human expert in a field. Knowledge engineering is the technology behind the creation of expert systems to assist with issues related to their programmed field of knowledge. Expert systems involve a large and expandable knowledge base integrated with a rules engine that specifies how to apply information

> in the knowledge base to each situation. The systems may also incorporate machine learning so that they can learn from experience in the same way that humans do. Expert systems are used in various fields including healthcare, customer service, financial services, manufacturing, and the law. Using algorithms to emulate the thought patterns of a subject matter expert, knowledge engineering tries to take on questions and issues as a human expert would. Looking at the structure of a task or decision, knowledge engineering studies how the conclusion is reached. A library of problem-solving methods and a body of collateral knowledge are used to approach the issue or question. The amount of collateral knowledge can be large. Depending on the task and the knowledge that is drawn on, the virtual expert may assist with troubleshooting, solving issues, assisting a human or acting as a virtual agent. Scientists originally attempted knowledge engineering by trying to emulate real experts. Using the virtual expert was supposed to get you the same answer as you would get from a human expert. This approach was called the transfer approach. However, the expertise that a specialist required to answer questions or respond to issues posed to it needed too much collateral knowledge: information that is not central to the given issue but still applied to make judgments. A surprising amount of collateral knowledge is required to enable analogous reasoning and nonlinear thought. Currently, a modelling approach is used where the same knowledge and process need not necessarily be used to reach the same conclusion for a given question or issue. Eventually, it is expected that knowledge engineering will produce a specialist that surpasses the abilities of its human counterparts.

Knowledge engineering can be applied to business rule development as it considers business processes, the tasks, agents and specifically the knowledge and reasoning used by the agents to carry out their functions.

One of the significant challenges with business rule optimisation is that the search space consists of all potential rule engines and items of knowledge that could be utilised for any business problem. This problem is like the challenge of expert systems where it is difficult to determine how decisions are made and what knowledge is used. However, we can sensibly delimit our ambitions around optimisation if we use the methods such as knowledge engineering to determine the knowledge base and the first cut of a rules engine. We then seek to improve by tuning the existing rules of adding new ones using optimisation and machine learning.

For example, in our case study on loan approval, we already know:

- The customer information that is used to make a decision
- Typical calculations – such as ratios - that are made using this information
- The extent of the knowledge base (all historical application, determinations, and outcomes)

Given the data we have available, the optimisation problem is one of identifying a subset of the information and a set of rules that maximises the profit function.

In the wider context, the problem can be decomposed into:

- Determining the extent of the knowledge base that could be used
- Creating the set of rules – that act on the knowledge base – that gives the optimal outcome

2.8 Job shop scheduling

Business processes can be viewed as a series of tasks that make up an activity. Enough of these tasks need to be completed in an order that results in the desired outcome, and task must be assigned to resources. In the manufacturing and engineering sector, there is a problem that consists of scheduling tasks (such as machining or assembly) that require specific resources (such as CNC machines or robotic welding) in such a way that costs or production are minimised or throughput is maximised. There are similarities to the allocation rules problem (the machines required for each workpiece in order).

According to Schauer (2013), the general JSS problem can be expressed as:

There are m machines $M_1, \ldots, M_m$ and k jobs $J_1, \ldots, J_k$ given. Each job J_i consists of n_i operations $O_{i,j}$ ($j \in 1, \ldots, n_i$). Every operation $O_{i,j}$ has to be performed $p_{i,j}$ time units on a dedicated machine $\mu_{i,j}$. Also, the order of the operations of every job is prescribed,

i.e. we have precedence constraints of the form of chains. $O_{i,j} \rightarrow O_{i,j+1}$ means that operation $O_{i,j}$ has to be finished before $O_{i,j+1}$ can start.

An essential consideration in JSS is the make-span. Make-span is defined as the total time taken from the beginning of processing to completion for any item. It is closely related to other criteria such as productivity and overall production capacity.

Flexible Job Shop Scheduling (FJSS) is an extension of JSS as it allows an operation to be processed by any machine out of a set of available machines, with allowable performance differences in the time taken to carry out each operation. As identified above, this is equivalent to a business process. We have elements of an allocation problem (choosing the machine there are multi-purpose machines, and choosing the order that these operations occur) where we want to maximise production, avoid under-utilisation and stay within the capacity limits of the machines.

JSS problems can be solved using integer programming, Baker et al. (2009). Subsequently, these problems have been solved by Genetic Algorithms (Agrawal, 2012) and Tabu Search, (Saidi-Mehrabad, 2007), and other methods that deal with an integer expression of the problem

FJSS has been extended and developed, and problems have been formulated and solved considering costs, start-up costs and the efficiencies of different machines, (Talbi et al., 2001). In this paper, he uses an evolutionary, multi-criteria optimisation approach.

FJSS considers the possibility of variability insofar as different jobs require different operations, but there are differences between FJSS and the research problem identified:

- The objective process all the jobs within a batch, or queue, at minimum time or cost
- All jobs are processed; they are not accepted or rejected

2.9 Queuing

Another critical concept in BPO is how to handle queues. Customers do not arrive at regular intervals, and the service proposition must consider queue length (the number of customers in the system) and waiting time (the time between arriving and completion of the service) (Bhat, 2015). Conversely, the organisation is interested in efficiency, and this is represented by the busy period, which is the time that a server is active. Note that we can consider any machine processing differently to human agents as:

- The marginal cost of processing is zero
- For all but the most complex transactions, time is virtually zero
- Any number of tasks can be carried out in parallel

Human agents are different; they take a finite time, cost money and are not scalable. Also, not all customers are happy to wait, and their tolerance for waiting is not uniform, so in any optimisation, we must consider waiting time. Customers may not join a long queue (balking) or may leave a queue (reneging) and depending on the circumstances, failure to serve a customer will have an adverse impact on an organisation.

These concepts are the domain of queuing theory and this will be required to ensure that any solution proposed to the business rule optimisation problem is workable. So, as well as looking to limit waiting time and queue length, we must also consider queue management, or at least the rules around queue management. Laguna (2013) considers these issues in some detail.

This presents an opportunity for an element of triage by the rules where a queue of potential customers can be managed such that we make optimal use of the human resources, subject to some safeguards that any customer will emerge from the queue within a given time, even if the optimisation algorithm might keep them at the back of the queue.

This would consist of:

- Identifying those customers who can pass through without human intervention
- Ordering the remainder so that the busy period for all levels of staff is maximised
- Making sure that no customer gets forgotten in the queue

2.10 Human Decision Making and Decision Support

2.10.1 Decision Making Process

When dealing with human agents we need to consider human decision making and the accuracy of human judgement.

In terms of the decision-making process, Harte (2001) summarises ideas that can be used to explain human decision making. Here we explore those methods that support a classification process.

The *linear additive strategy* is a linear combination of the attributes, with suitably chosen weights, and decisions are based on the overall sum.

With the *conjunctive strategy* the attribute values are compared to some thresholds for each attribute. If the attribute values of an alternative do not meet these thresholds, the alternative is rejected. If this strategy does not result in the desired number of remaining alternatives, the thresholds must be adjusted, and the procedure has to be repeated. Effectively, we require all attributes to meet the threshold assigned to it.

If the *disjunctive strategy* is applied, the alternatives are also evaluated one by one by comparing the attribute values with some thresholds. But, contrary to evaluation using the conjunctive strategy, an alternative is selected if at least one of the attribute values exceeds or equals the threshold. Here we require only one attribute to meet the threshold assigned to it.

The linear additive strategy is very commonly used, in credit scoring, for example, and is easily suited to categorisation. But it requires weights and limits to be determined.

The conjunctive and disjunctive strategies may have threshold values for each attribute. As such, they can be used for categorisation. The conjunctive strategy (testing each attribute against an attribute specific limit) is typically employed for screening by a computer.

2.10.2 The LENS Model

According to Kaufmann (2013), the LENS model identifies multiple components of judgment accuracy. In a typical LENS model study, a 'judge' must make decisions based on different pieces of information ('cues'). Judgmental achievement is measured by the extent to which the judge's judgment matches (i.e., correlates) with an indicator of the actual outcome or situation ('criterion'). The LENS model is the basis for the LENS model equation that mathematically describes judgmental achievement (r_a, i.e., the correlation between a person's judgments and a criterion) in terms of four components. Namely, the judgmental achievement is equal to a linear knowledge term (G) multiplied by task predictability term (R_e) term multiplied by a consistency term (R_s) plus a non-linear knowledge term (C).

The linear knowledge component (G) refers to the correlation between the predicted human judgment and the predicted criterion. Task predictability (Re) refers to the multiple correlations of the cues with the criterion, Consistency (R_s) refers to the reliability of judgments, that is, the extent to which a judge reliably reaches the same decision based on the same pieces of information. The non-linear knowledge component (C) represents the correlation between the variance not captured by the environmental predictability component or the consistency component.

Previous research has revealed that the non-linear knowledge component is generally quite small. The definitions of the single components are:

r_a = the achievement index (i.e., the correlation between a person's judgments and the criterion),

R_e= the task predictability index (i.e., the multiple correlations of the cues with the criterion),

R_s =consistency (i.e., the multiple correlations of the cues with a judge's estimate),

G = a knowledge index that reflects achievement (i.e., the correlation between the predicted levels of the criterion and the predicted judgments), and

C = an un-modelled knowledge component that signifies the correlation between the variance not captured by the environmental predictability component or the consistency component (i.e., the correlation between the residuals from the above predictions).

Figure 1 shows the components of the LENS model with reality on the left-hand side and judgements on the right.

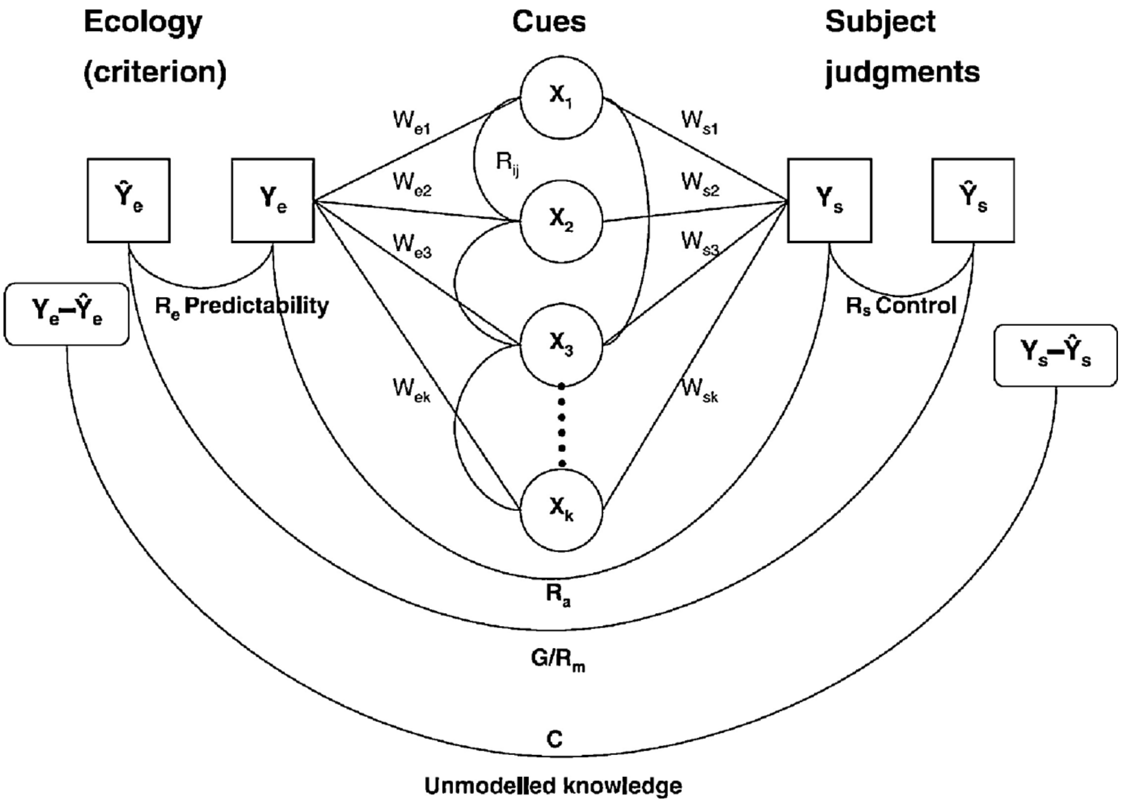

Figure 1 LENS Model (Karelaia et al, 2008)

The LME is:

$$r_a = GR_sR_e + C\sqrt{1-R_s^2}\sqrt{1-R_e^2}$$

In essence, if we ignore C as generally small, the model determines that achievement (accuracy) of judgement increases linearly with knowledge, predictability and consistency.

The LENS model is useful in modelling human decision making, but when we introduce business rules, we have another dimension. We potentially have three outcomes: the human decision, the rules decision and real outcome. We can replace the human decision maker by the rules and have a LENS model that consists of rules decision and outcomes, or better still, have add an extra dimension and have all 3 in one model. The question then is to determine which decision to choose. Whilst the LENS model gives us information about the overall accuracy of either decision on a specific case, it does not give any indication on a case-by-case basis. As such, we must extend the formulation such that:

i. We add the business rules as an extra dimension to the model and create a 3-Dimensional LENS mode

ii. We include a case-by-case calculation of the accuracy of the human decision maker and the rules to facilitate choice between the two

As final point, in the case where the decision is binary or integer, linear regression may not be the most effective method of modelling and we may need to consider other methods.

2.10.3 Decision Support Systems (DSS)

Computers can be used to help or support the human decision-making process. These are primarily concerned with data into actionable information (extraction, synthesis, calculation, and presentation). They can also assist human decision makers with further data, analysis and 'what-happens-if' analyses. For example, taking a company's order book and sales prospects across different divisions, combining that with a projection of costs to calculate profit and revenue projections under different scenarios. The human decision maker can then use their judgment to decide on the most likely scenario(s) and associated outcome(s) and make decisions accordingly.

According to Sprague (1980), the characteristics of a DSS are:

i. DSS tends to be aimed at the less well structured, underspecified problem that upper level managers typically face.
ii. DSS attempts to combine the use of models or analytic techniques with traditional data access and retrieval functions.
iii. DSS specifically focuses on features which make them easy to use by non-computer-proficient people in an interactive mode; and
iv. DSS emphasizes flexibility and adaptability to accommodate changes in the environment and the decision-making approach of the user.

Decision support is a mature technology and has become business as usual in many sectors (Alter, 2004). More sophisticated systems support multiple criteria decisions (Wallenius, 1992) that involve financial, societal, and environmental considerations.

Decision support systems use a knowledge base (as in 2.7) and models to process situational information, forecasts or scenarios (inputs) and create outputs, such recommendations or options with implications such as costs, resources, or customer impacts, for example.

There is a degree of overlap between business rules and a DSS. For example:

- A DSS may consist wholly, or in part, of rules
- Business rules may provide inputs to, or act on the outputs of, a DSS. For example, there may be a rule that requires decisions with specific implications to be approved by more senior staff. Or there may be rules that mandate or limit the use of DSS certain situations
- DSS may provide qualitative or quantitative information on the reliability or accuracy of its outputs or advice. However, there is no indication that current DSS platforms include models of the human decision maker, or the relative merits of human versus DSS decisions. These models may be a useful capability and use some of the ideas from this thesis.
- DSS may incorporate optimisation technology (Wallace, 2020). This is typically situational optimisation (2.4) as in many other optimisation applications.

There is nothing to prevent optimisation of the rules within a DSS. The value of the economic or other outcomes over a range of historical or expected situations can be maximised by choosing the best set of rules. On the contrary, a collection of optimised business rules that incorporates the human decision maker could be viewed as a DSS.

2.11 Cost of Information

There is a cost of obtaining and processing information. This cost has been incorporated in extensions to decision tree learning (Lomax et al., 2013). There is also a cost in terms of transaction abandonment (or cart abandonment with on-line transactions) where there is a tendency for customers to give up as the amount of information requested, or the time taken, increases. Most of the research focuses on causes (Xu et al., 2015) and prevention (Zimmerman et al., 2016), and concentrates on the overall rate (van der Geest et al., 2016).

To support our analysis of the process, we require to model the relationship between the time or amount of information requested and the proportion of customers that give up or abandon the transaction. This relationship is an area that has been neglected.

2.12 Parallels with Manufacturing and other Non-Service Industry

As previously stated, manufacturing has typically focussed in business processes rather than business rules per se. Take up of optimisation techniques is widespread with critical elements (Lawler et al., 1993):

2.12.1 Long- and short-term demand forecasting

Demand forecasting is a vital element of any forward-looking optimisation of decision support. Techniques have been developed to combine stochastic and deterministic variations in demand that could be applied to the services sector (Aburto et al., 2017).

2.12.2 Optimisation

There are three basic levels (Bilgen et al, 2004).

- Strategic optimisation: process design and investment in resources.

  The key issue here is to match equipment capacity to the anticipated long-term demand. This level includes investment and capacity planning.

- Tactical optimisation: production planning and resource allocation.

  At this stage, the rates of production are decided, and production is allocated to the most appropriate, efficient capital and human resources.

- Operational optimisation: production scheduling.

  This deals with short-term variations in demand and schedules production to meet current and short-term anticipated demand.

The provision of services consists of a series of related tasks that are carried out to satisfy the needs of a customer (or customers). We can draw some similarities between these issues and the services sector. There are even some more profound similarities. For example, in manufacturing, there is the so-called lot-sizing problem which is to determine the optimal production run considering start-up/changeover cost and the cost of holding stock. If lots are too small, then start-up/changeover costs are too high; too large and more stock is produced, and the inventory cost is too high. There is a similar problem in the services sector where the same customer service staff can serve customers, act in the call centre and process paperwork in the back office. Demands for all three can vary and to be efficient staff allocation is based on demand or expected demand. There are also changeover costs as staff shut down one activity and start another. This problem is like lot-sizing; shorter periods enable maximum flexibility and higher overall efficiency but make them too short, and changeover costs become unacceptably high.

The transport problem, which is a case of disruption management, is typically solved by using optimisation techniques to determine a new schedule. This process is complicated and expensive and, like any optimisation problem, heavily dependent on the availability of large amounts of accurate data. Rule-based approaches have been used in timetable generation, particularly for conflict resolution, and these could be developed further to inform the rules that transport system controllers (implicitly) use to resolve disruptions to create greater consistency. The simple example could be used as a way to manage delays in the system.

2.13 Research gap

Business rules can be viewed as constraints around a process (such as health and safety legislation, company rules, etc.) but they are also embedded in activities and tasks (what to do next with an insurance claim form with a particular set of attributes). A typical BPO approach would be to model the insurance claim process would have a node called assess the claim and maybe three outputs: reject, refer or approve. A further assumption would be the proportion of claims that progress to each of the three nodes. This process can be optimised by having only the essential tasks in each activity, having the minimum number of claims processors and underwriters so that claims are processed in the required time, queues are manageable, and costs are contained. But a significant consideration in insurance is to pay only those claims that are valid, that is; not to pay when you don’t have to/more than you have to, and not to reject valid claims that would upset customers and damage the brand. Ideally, every claim would be scrutinised in detail by an underwriter, but this is overkill and too expensive. So, we need rules that determined which direction *each* claim takes through the system. To properly optimise the process, we must optimise these rules.

BRO, therefore, can be considered an essential part of BPO whenever there is variation in inputs or externalities that impact on costs or profits. The simple example, above, demonstrates that it is not always possible to optimise the process first and then the rules. BPO has to make assumptions about the proportion of claims that require underwriting; optimising the rules may results in a different proportion, invalidating one the key assumptions.

Waiting time (queueing and servicing time) is another consideration. The rules optimisation problem must include waiting time and the cost and availability of resources. A subsidiary issue is how to manage a queue; what rules do we apply when selecting the next customer or case to service.

Several issues suggest that there is a significant research gap that needs to be bridged if we are to address the business rules optimisation problem that has been identified. The current state of the art is:

i. Optimising business processes without considering rules

ii. Rules that seek to emulate or replace the human expert

iii. Scheduling manufacturing tasks to minimise make-span

iv. Queuing without considering business rules

There is a requirement to bring these together to maximise the opportunity to optimise rules around processes, classification AND queue management, and there are some other issues around the application to the services sector.

While consistency is valuable in the services sector, it is a matter of experience that all customers are different, and there are some types of interactions that are missing from the manufacturing scenario. For example, we have complex tasks such as:

i. choosing a product or service for a customer

ii. qualifying and accepting a customer for a loan or insurance policy

iii. dealing with complaints

In each of these, as in every business activity, cost and time are essential. But so is getting the right result, such as accepting all the customers we should for insurance cover at a reasonable price; spending enough time and money (but not too much) to keep or win back an unhappy customer.

Table 3 summarises the known gaps in the relevant fields of research.

Research Area Requirement	Business Process Optimisation	Business Rules	Flexible Job Shop Scheduling
Considers how much information (input) is required	No	No	No
Considers decisions about each customer/task	No	Yes	No
Considers how individual customer cases flow through a process	No	Yes	No
Considers the quality and cost of decisions	No	No	No
Considers queues and their impact	Yes	No	Yes
Manages queues	Yes	No	Yes
Optimises queue management	No	No	Yes
Considers judgement and levels of expertise	No	No	No

Table 3 Research Gaps

There is an additional optimisation problem that can be posed. Once we have optimised the rules (for a population of customers), we can still optimise other issues dynamically. These include:

For a given resource allocation, we develop ways to adjust the rules, dynamically, so that we will make the best use of them.

Methods to manage the queue of cases such that we have a sufficiently large sample to make meaningful rule adjustments without causing undue waiting time.

2.14 Summary

There are two ways to view previous research:

i. The research has addressed some of our research questions

ii. The research provides useful tools to address the broader questions

BPO addresses the wider question of creating optimal business processes, but largely ignores business rules. When it does allow for them, it models their impact but does not identify rules as a candidate for optimisation.

Business rules research covers many aspects from creation, operation, maintenance and impact. About impact, there is a recognition that rules impact key performance indicators, and they can be used to optimise research allocation. But there is no attempt to optimise the expected value of profit, revenue or some such other financial measure. While there is the consideration of allocation of work to humans, it is not in the context of the comparative advantage of the rules or the human making the decision.

Job shop scheduling, or FJSS, is like the allocation rules problem but is largely driven by efficiency or time improvement; quality of decision relating to the handling of different tasks is not a consideration.

The LENS model was developed to understand human judgement and has been widely applied. In our research we need to extend it to include the decisions made by human experts and rule systems.

It is interesting to note that human decision making can be considered as a set of logical tests – such as comparing attributes to limits – and weighting the attributes and testing the sum (or score) against a certain threshold. At the same time, the LENS model using linear regression. Both methods have the advantage of being easy to understand. As such, building rules that consist of logical tests or regression is a sensible starting point for application. These sorts of rules are also common within commercial and industrial implementations of business rules.

In conclusion, each of the areas above provides useful building blocks and tools to address the current research questions, but of themselves, do not address the issues directly.

3 SIMPLE EXAMPLES

3.1 Introduction

In this section, we look at some (deliberately simplified) examples of service business processes that are (or could be) controlled by business rules. We choose an objective function in each instance and an example rule and show how the choice of a parameter, such as an upper or lower limit, is an optimisation problem. We also relate these simple examples to some of the research questions and hypotheses. The objective of this section is to show that, even in the most simplified of situations, there are opportunities and potential benefits of business rule optimisation.

The service business processes are credit approval (the case study), debt recovery and transport management. In the first two examples, the objective function is related to profit and can include the option to refer the case to a human decision maker. In the third example, the objective is customer satisfaction. As such, they illustrate the point that optimisation can deliver more than just financial benefits.

The essential learning is the importance of expressing the expected value of profit or revenue (or indeed, customer satisfaction) as a function of the parameters within the rules. This expected value is the objective function of the optimisation problem, and the parameters are the optimisation variables.

3.2 Business rules in the services sector

Business rules in the services sector are about guiding decisions. As information is processed, decisions are made, and further information is generated. For example, a customer enters a bank and asks for a loan. Information is gathered about the customer and what they want, and the result of analysing that information other information is generated, e.g., the customer is accepted for a loan of a particular value with specific repayment profile, etc. The correct application of rules enables consistency and management by exception. For example, a skilled and experienced person is not required to be present at every consultation; he/she only needs to engage where their judgment or approval is essential.

3.3 Research Questions & Hypotheses

We have created simple problems to illustrate some of the research questions and one hypothesis. These also set the scene for the rest of the thesis,

We have:

RQ1 How is the optimisation problem to be defined?

RQ2 How should the rules be optimised?

RQ5 How can we incorporate the human expert to best effect?

H1 The underlying hypothesis is that it is possible to optimise any set of business rules in the sense that they give the best results over a defined range of situations (either determined by analysis of historical data or forecasts).

In these examples, we illustrate that if we have information on the following,

- Gains from making a correct decision
- Losses from making an incorrect decision
- Probabilities that decisions are right and outcomes are good

we can create a set of rules that maximise the expected value.

3.4 Loan Application Example

In many situations in the service sector, actions are based on information. This information is either already available or can be elicited from a customer as part of the interaction. A simple example is requesting a loan. The customer can choose from several options (length, secured or unsecured, interest only period, etc.). On initial application, some decisions can be automatically made either by an unskilled worker or the back end of a website using business rules. The possible outcomes are decline, refer (to a more skilled and experienced staff member) or approve. The business rules will use information (such as the customer's income and credit history), combinations of information (such as the ratio of loan to the customer's income) to decide in which category the customer should be. Figure 3 is a simple diagram which considers two factors: loan amount and customer income. The rules have an impact on the business in several ways. Consider the decline/refer interface. If the rule is too conservative, then more customers are turned down leading to a loss of business and customer unhappiness; if the rules are more aggressive, then more customers are referred (which costs money) and may ultimately be declined. Similar considerations apply to the refer/accept interface. So even in this simple case, there are trade-offs and an optimisation problem to be solved.

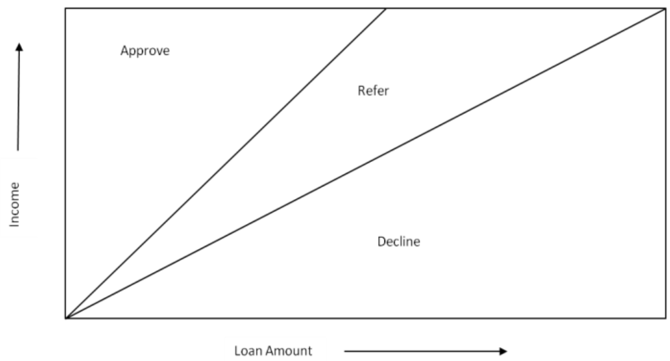

Figure 2 BUSINESS RULES FOR LOAN APPLICATIONS

The business rule illustrated in Figure 1 may be stated as follows:

IF (loan/income) < x_1 (accept limit) THEN approve = true

ELSE IF (loan/income) < x_2 (refer limit) THEN refer = true

ELSE decline = true

To treat the loan acceptance problem as an optimisation problem, we need some further information such as:

- the probability of default as a function of loan value/income (x)
- the losses associated with default. For example, loss of interest or capital and cost of foreclosure
- the additional cost of referring an application
- the value of a customer to the organisation, that is, expected profit
- the distribution of income/loan value across all customers

Let us assume (and these are straightforward assumptions for the sake of illustrating the point and obtaining an analytic solution) that:

- Cost of default per customer = D
- Cost of referral per customer = R
- Potential value of customer = V

- x is the loan/income ratio, and x is distributed uniformly (for the sake of simplicity) over [0, 10]. For a mortgage, the loan is generally greater than income and a typical value is between 3 and 5
- P(default) = x^2 / 100 (probability of default increases to unity with x = 10). Note, in practice the probability of default would be more complex (and be related to the normal distribution, for example), but we choose this for computational convenience.
- Referral reduces the rate of default by 75% and rejects 50% of applications. Again, this is a simplification; referral will reject applicants, and the remainder should be less likely to default.
- The objective is to maximise revenue.

We have to choose parameters, x_1 and x_2, for the interfaces between rejection and referral. The cost function is made up as follows:

First, we calculate the expected value of customers automatically accepted (since we are assuming a uniform distribution). This is the fraction with loan to income ratio less than x_1 multiplied by the value

$$v_A = V.{}^{x_1}/_{10}$$

Now we calculate the expected value of customers accepted after a referral process that rejects 50%. This is the fraction of customers with a loan to income ratio between x_1 and x_2 multiplied by the net value $V - R$ and the 50% acceptance rate.

$$v_B = (V - R).{}^1/_2.{}^{(x_2 - x_1)}/_{10}$$

Finally, we calculate the expected cost of defaults (based on our assumed default probability relationship and the reduction in default rate of the cases referred). This is the default cost D multiplied by the expected faction of defaults between 0 and x_1 using the original probability of x^2 / 100 and the expected fraction of defaults between x_1 and x_2 using the probability that has been reduced by 75% (x^2 / 400).

$$c = D\left\{\int_0^{x_1} x^2/100\,dx + \int_{x_1}^{x_2} {}^{x^2}/_{400}\,dx\right\}$$

We then calculate total value minus costs

$$F = v_A + v_b - c$$

$$= V.{}^{x_1}/_{10} + v_B + (V-R).{}^{1}/_{2}.{}^{(x_2-x_1)}/_{10}$$
$$- D\left\{\int_0^{x_1} x^2/100\,dx + \int_{x_1}^{x_2} {}^{x^2}/_{400}\,dx\right\}$$

$$\frac{DF}{dx_1} = \frac{V}{10} - \frac{V-R}{20} - D(\frac{x_1^2}{100} - \frac{x_1^2}{400})$$

$$\frac{DF}{dx_2} = \frac{V-R}{20} - Dx_2^2/400$$

With the derivatives set to zero, the location of a stationary point is given by the equations:

$$x_1^2 = {}^{20(V+R)}/_{3D}$$
$$x_2^2 = {}^{20(V-R)}/_{D}$$

Intuitively these make sense. If the value of *D* increases, both parameter values go down; if V increases, both parameter values go up, and if R increases the gap between them decreases to the point where there is no gap when R = V/2. With V = 15, R = 1 and D = 10, the maximum value is achieved when $x_1 = 3.27$ and $x_1 = 5.29$

This simple example illustrates the general approach of finding the measure of system performance as a function of the parameters within the rules, then finding the best values for these parameters.

3.5 Debt Waiver Example

Another example that creates an optimisation problem is one of debt waiver. Data usually is available on the recovery rates of debts of various sizes and the cost of their recovery (administration, legal fees, etc.), at the intersection of this line and the recovery cost, we can read off the value of the smallest debt that is economical to enforce. This minimum level is then the point, below which, debts would not be pursued; they would be waived.

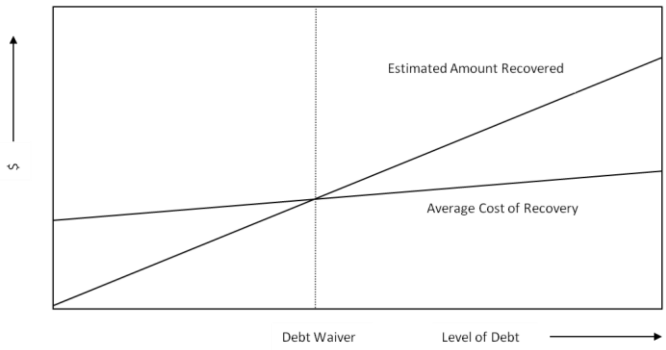

Figure 3 DEBT WAIVER

We can also incorporate the human decision maker in this example by adding an extra step. If the estimated amount recovered is higher than the cost of recovery by only a small margin, the case may be reviewed (by a human expert) before we proceed. The criterion for review would be determined by the difference between the probability that the debt would not be recovered (even if we pursued it) multiplied by the cost of recovery compared to the cost of a review. This criterion implicitly assumes that the expert is always right. Still, we can allow for that, and consider the probability that the expert is correct, or incorrect, in his or her judgement.

3.6 Transport Rules

It is quite common in transportation systems to have rules governing disruptions, e.g., a train is late or disabled, or conflict, e.g., two trains requiring the same track at the same time (D'Ariano, 2004). A typical problem is what to do when train *A* is late, and there is a connection to another train, *B*. Typically *B* waits for up to a maximum cut-off (c) for *A* where $0 < c < f$, where f is the time until the departure of the next train.

The trade-off in choosing c is the inconvenience to passengers in train *B*, for waiting, against the inconvenience of passengers in train *A* for having to wait for the next train.

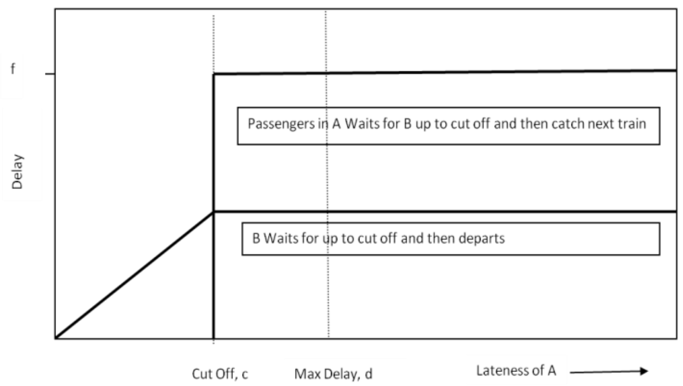

Figure 4 Transport Rules

If we wish to be fair to passengers in A and B, then the best value for c depends on the distribution of delays to A and the interval between trains, f. If we assume that the distribution of delays is uniform between [0, d], we have the following equations.

The expected additional wait for passengers in A when they have to wait for the next train is the probability that the delay exceeds cut-off times the wait for the next train.

Expected (additional delay *A*) = $f\,.(d-c)/d$

The probability that the delay is up to the cut off is c/d, and the expected value of such a delay is $c/2$. Therefore:

Expected (additional delay *B*) = ${}^{1}\!/_{2}\,.{}^{c^2}\!/_{d}$

For equal inconvenience, we have:

$$(d-c).f = {}^{c^2}\!/_{2} \Rightarrow$$

$$c^2 + 2cd - df = 0$$

This is a quadratic equation and can be solved for c. For example, when d = 10 mins and *f* = 20 mins, *c* = 8.33 mins, close to the maximum delay anticipated.

Also, there may be benefits to considering the number of passengers in trains *A* and *B* to make the rule genuinely fair.

In this example, the role of the human decision maker is somewhat different. It may be limited to making a decision only when the ratio of *A* to *B* is outside some set limits and requires an element of judgement.

3.7 Other Examples

There are many other examples of rules that affect the performance of an organisation. For example, in public services, we have the concept of triage where cases come in, and the problem is to determine the urgent cases. This idea is particularly important in health and social service, such as child protection. For example:

- In health, there are limited resources, and the arrival of cases is not uniform. As such, if each case were attended to as they arrive (first come, first served), the waiting time would vary – sometimes significantly. Extended wait times could be harmful or indeed fatal for some cases.
- Child protection and social services (such as domestic violence) are similar. Cases that are considered to be high risk need identification quickly so that that action can be taken promptly. Otherwise, we create an unacceptable probability of harm, which is related to both risk and time exposed to it.

But a good definition of urgent requires a rule and parameters. These can be determined from historical data (or forecasts) of queueing times and the proportion of cases that are exposed to risk by waiting. Still, risk reduction must be balanced against the additional resources required to triage each case, even though a large proportion may go back into the queue and require processing later. Also, we must limit the number of cases that are classified as urgent if that would overwhelm the available resources.

There is also the issue of fluctuating demand. Rules must remain relevant even when demand is changing.

3.7.1 The Additional Optimisation Challenge

As mentioned above, work has been done in business process optimisation. That involves the structure and order of tasks to minimise time, minimise cost or maximise throughput. The critical feature in the examples above is probability. In the first, it is the probability that a customer, if pursued, pays up. In the second, it is the probability of default, and in the third, it is the probability of delay. Where there is a range of customers, or requirements, or external factors that impact the business problem needs to be considered in the optimisation of rules. Later, in

Chapter 5, we consider optimisation of expected values, and in Chapter 8, we apply this to a real problem with real data.

3.7.2 Real Life Problems

Real life problems are more complicated. For example:

In the loan example, there will be more than one criterion. Credit history or value of the security may also be considered (FDCI, 2007). The interest rate may vary with the customer's credit rating or size of the loan, affecting affordability.

With debt waiver, there may be additional rules (also subject to optimisation) that govern recovery and affect the cost. For example, it is not unusual for debts above a specific size to be subject to mandatory legal action

In transport, there are additional complications. People travel in both directions, there are knock-on effects of delays around the network, and the concept of fairness may be difficult to sell to customers (Visentini, 2014).

The order in which rules are applied is also important. For example, servicing customers costs money so finding out sooner rather than later that a customer does not qualify for a loan is an advantage

The rules that need to be applied to minimise cost or maximise profit may not be evident or apparent. If we have not explored every possible rule, in every possible combination, then we cannot be sure that we have the best set. However, what can do is optimise the following variables that make up the CRA rules:

- The choice of attributes that we use
- Choice of operators and constants in any *calculations*
- The choice of logical tests, and their application to the attributes (*reasoning*)
- The *allocation* decisions

3.7.3 Generalisation

These examples do illustrate some crucial issues for rule optimisation:

- Modelling the impact of (changing) a rule on revenue, costs or customer experience

- Optimise the expected value of the objective function. This value can be based on probability distributions (as in loan approval) or historical information (as in debt waiver).
- The existence of, and a solution for, an optimisation problem, even simple examples
- The concept of rules that can decide or that can refer, and the cost and impact making the decision using a rule versus the consequences of referral, a different quality of decision and associated cost

3.8 Summary

We have presented some simplified (but real world) examples where the choice of business rules has a real impact on the financial or customer service performance of an organisation. In all cases, there is an optimisation problem where we are maximising an expected value; and probability is a key element. The rules must work over a range of inputs and responses, and when we optimise them, it must include probability distributions. The example problems relate to some of the research questions, and we have shown that in each case:

- There is an optimisation problem (RQ1)
- There is a solution (RQ2)
- We can optimise a set of rules (RH1)

Also, in the case of the credit approval problem:

- We can incorporate the idea of referral to a human expert that has an additional cost and the ability to decide between applicants (as in the loan approval example)

4 SERVICE BUSINESS PROCESSES

4.1 Introduction

In this chapter, we identify a wide range of service business processes that would use CRA rules, and to which our proposed problem statement and solution methodology will apply. We recognise the opportunities for rules, executed by a machine, and the judgement of human experts to work together in a framework and determine that four problems arise when considering business rules and human decision makers. These are:

i. The optimal quantity of information to be collected, and in what order (the feature selection problem)

ii. The CRA problem – finding that set of rules that makes calculations, reasons and then direct the cases to the machine or human expert for the best outcome

iii. Resource determination – methods to determine the right number of experts for any given caseload and case-mix

iv. Operational adjustment – methods to dynamically modify the rules when the caseload or case-mix changes

We first look at a range of service business processes that make decisions based on information provided by individuals (such as customers, patients and claimants). Based on this, we create a set of key concepts and a model of the process that includes the benefits of making correct decisions, the costs of making bad decisions and the cost of gathering and processing information.

4.2 Example Business Processes

We begin by analysing the characteristics of service sector processes that would use CRA rules. These include:

- Loan/insurance application

- Insurance claim
- Making an investment
- Entering a contract
- Deciding on a merger or acquisition
- Staff recruitment
- Diagnosis
- Fault finding
- Sales
- Child protection
- Fraud detection
- Visa application

We need to find a way to express the essential elements of each process so that we can build a model process that will serve as a framework for business rules optimisation.

There are common elements:

4.2.1 Actors

i. The *applicant* (or promoter). This is the person (or organisation) who (that) provides the information in the first place. He/she/it may also be responsible for providing further information.

ii. The *decision maker*(s), which are essentially agents (machines and people) that follow rules and processes, and make judgements

iii. The *agents* involved. An underlying assumption here is that agents may be machines or humans. In machines we can implement any form of AI; in humans we must characterise their decision making. Still, we cannot replace it with a machine (otherwise we would, on the grounds of cost saving and consistency).

4.2.2 Financial

i. The *upside*: the benefit (or saving) when a decision to proceed results in a good outcome. For example, the other party keeps to the terms of the contract; customers pay on time, in full; customers do not defraud, etc. This could also be expressed in non-monetary terms, such as a person gets better or returns to work.
ii. The *risk*, which reflects the reality that at the point of making a decision, we don't know how things will turn out
iii. The *downside,* which is the cost of a decision to proceed that results in a bad outcome.
iv. *Information cost.* For providing or acquiring information, that can be incurred by the applicant, the decision makers, or both.
v. The *processing cost*, which is generally incurred by the decision makers in paying for people to read and assess the information.

4.2.3 Information

i. The *information* set that supports the decision. This is made up of attributes that fall into categories or are ordered (real, continuous). It can be safely assumed that this can be digitised and machine readable, and it is not required that the information is completely accurate.
ii. *Historical data* on decisions, and good and bad outcomes, which can be used as a training set or to determine the joint probability distribution of good and bad outcomes. This can also be used to characterise the ability of humans.

4.2.4 The Decision

i. The *decision* itself, which is often ultimately binary (accept or reject) but the rules themselves determine choices such as accept, reject, refer and request more information. Some business decisions require clear choices to be made. Medical diagnosis and fault finding requires a clinician (or technician) to choose tests (or investigations) based on the symptoms. There is another slant which is to accept/proceed with a condition. For example, we would buy at another price; we would hire at another salary. This is a dual problem as far as we find the highest purchase price/salary that results in a decision to accept (see Chapter 16 for an explanation).

Tables 4-7 below show service business processes, the characteristics of which have been elicited from industry professionals based on their knowledge and experience. We have performed a process mapping (Hunt, 1996) on these processes. The objective is to understand the similarities between the case study example (loan application) and a range of other business processes that may appear different but do, in fact, have the same characteristics.

Service Business Process	Loan/Insurance Application	Insurance Claim	Purchase or Investment Decision
Applicant	Customer	Policy holder	Internal champion
Decision Maker	Underwriting	Loss adjuster	Management
Upside	Margin from new business	Pay claim and make customer happy	Financial benefit of making purchase
Risk	Customer does not behave as expected	Policy holder does not behave as required or expected	Purchase does not deliver on expectations
Downside	Cost of default or fraud	Paying unnecessary claim	Cost of item/service
Decision	Accept/reject	Pay/decline	Proceed/don't proceed
Agents	Customer service Agent Underwriter	Customer service Agent Loss adjuster	Purchasing Agent Purchasing executive
Information	Application form	Claim form	Business case
Information cost	Customer time Customer may give up	Customer may not provide information diligently	Time to prepare a case Champion may be put off/give up
Processing cost	Reading application form	Reading claim form	Reading business case
Historical data	Past cases of customers accepted with outcomes	Claims made and accepted/rejected	History of purchase decisions with outcomes
Source	Head of Risk Secured Lending ANZ Bank	Chief Risk Officer AIA Insurance Australia	Partner Access Capital Advisers

Table 4 Loan, Insurance and Purchasing

Service Business Process	Entering into a Contract	Merger or Acquisition	Recruitment
Applicant	Salesman	Internal champion	Interviewee
Decision maker	Contract approval	Board	Interview panel
Upside	Proceeds of sale	Increased revenue and profit	More effective team
Risk	Contract fails to deliver as expected	M&A fails to perform as expected	Employee does not perform as expected
Downside	Up to full contract value	Up to full acquisition cost	Salary and training costs
Decision	Proceed/do not proceed	Proceed/do not proceed	Hire/do not hire
Agents	Contract manager Sales manager	CEO Board	HR Senior manager
Information	Contract terms and conditions: expected margin	Due diligence Market information	Application form Interview
Information cost	Preparing summary of key terms Opportunity cost may be too high	Preparing summary of key terms Opportunity cost may be too high	Too many questions may put off potential applicants
Processing cost	Reading briefing documents	Reading briefing documents	Reading application Conducting interviews
Historical data	Past performance of investments	Past performance of M&A	Experience with hiring new employees
Source	Head of Procurement Hitachi	Chairman Australian Venture Capital Association	Manager Hays Specialist Recruitment

Table 5 Investment, M&A and Recruitment

Service Business Process	Medical Diagnosis	Fault Finding	Sales
Applicant	Patient	Equipment or system	Salesman
Decision maker	Health provider	Maintenance company	Buyer
Upside	Patient lives a full life	Machine works well	Make a sale
Risk	Patient does not respond as expected	Fault cannot be fixed	Product offered does not meet customer needs or is not competitive
Downside	Patient lives an impaired life/dies	Machine inefficient or ineffective	Sale is not made
Decision	Patient has condition/doesn't have condition.	Fault yes/no	Buy/not buy
Agents	Doctor Specialist	Diagnostic computer Technician Engineer	Pricing software/price list Salesman Sales manager
Information	Symptoms Answers to questions Tests	Sounds Smells Visuals Measurements	Customer information Dialogue Body language
Information cost	Patient/clinician time Cost of tests Side effects	Technician/engineer time	Time to meet customer Creating proposals
Processing cost	Clinician time	Technician/engineer time	Meetings Reading proposals
Historical data	Patient/hospital records	Manufacturers data Machine history	Sales records Dealings with potential customer
Source	Chief Medical Officer Monash Health	Asset Manager Goulburn Valley Water	Manager Hays Sales and Marketing

Table 6 Medical Diagnosis, Fault Finding and Sales

Service Business Process	Child Protection	Fraud Detection	Border Control
Applicant	Social worker (on behalf of the child)	Transaction processor	Visa applicant
Decision maker	Child protection agency	Service provider	Immigration authority
Upside	Child is safe	Fraud is prevented	Visa is issued to traveller with no adverse consequence
Risk	Child may be subject to harm	Fraud or intention to defraud	Traveller intends to break the rules
Downside	Child is harmed	Fraud occurs	Traveller breaks the rules
Decision	Further action?	Is there a fraud?	Issue visa or refuse
Agents	Social worker Allied health professionals	Computer program Fraud assessor	Computer program Immigration officials
Information	Case notes Interviews	Transactions	Application form
Information cost	Visits Interviews	Automatically generated	Interviews
Processing cost	Meetings Case conferences	Checking	Processing applications
Historical data	Previous cases	Payment history Buying patterns	Case histories
Source	Research Director Dept Education and Early Childhood Development Victoria	EGM Privacy, Identity and Cyber Commonwealth Bank Australia	Director Dept Immigration and Citizenship Australian Government

Table 7 Child Protection, Fraud and Border Protection

Looking at this list, we can draw some conclusions:

i. The final decision may be binary or integer. A decision to proceed, or not, is binary. A diagnosis may be a choice from several options.

ii. Integer decisions are still within the scope of CRA rules as the rules can consist of multiple paths and reach numerous conclusions.

iii. It is difficult to define an objective function when the upside and downside are not in the same units. For example, upside may be health, downside maybe illness but diagnosis costs are in $.

iv. The practical difficulties that occur when primary inputs are not digital or easily digitised. These are particularly important with information collection. For example, responses to open ended questions are more difficult to process and can be replaced by options or closed questions.

Choosing that set of business rules to maximise the profit of an organisation that is processing cases, with each requiring a decision, needs an understanding of the business and its business processes. We assume that the cases are processed manually by staff of different costs levels, and levels of expertise. The result of the optimisation will include the rules, anticipated profit and resource profile of the organisation.

In the case study, we confine our attention to business problems where the decision is binary (that is, at the end of the process, there are only two options). There are, however, many paths that can be taken, for example decisions made by different means or actors. This is for two key reasons:

- The description and analysis of the problem is more straightforward yet extensible by virtue of (i) and (ii) above
- The data set that we have is one where the final decisions are binary so that we can test our ideas fully on this data.

4.3 Key Concepts

We have focussed on CRA business rules that are commonly used in the services sector to make decisions or process cases. We have selected a case study in loan approval to realise the problem definition and use real data to demonstrate practicality and benefits.

To better understand the business rule optimisation problem, we give an overview of important concepts:

i. We have an organisation that is processing *cases* that require decisions to be made around *acceptance* and *rejection*. A set of information characterises each case. Acceptance enables the organisation to enter some sort of arrangement that includes the opportunity to make a *profit* or a *loss*. At the time of entering the arrangement, it is not known whether a *profit* or a *loss* will result. For example, an insurance company can accept a customer that will pay premiums, but that may also make a claim. An engineering contractor may approve a design that leads to payment, but that design may have a flaw that results in liability later.

ii. While the overall *process* may appear straightforward - a case comes in and is either rejected or accepted - there are different *actors* in the process and different *paths* through the process. The routes through the process are determined by *business rules* and by the actors applying both rules and *judgment*. The rules and judgments are applied to *information* provided with a case which may be *complete* or *incomplete*. For example, an insurance company will have a website that applies some rules and underwriters who can use their judgment (again within an area determined by the rules). In an engineering company, the sign-off level for a design may be determined by rules acting on the size or nature of a contract.

iii. The optimisation questions concern *design* of the process. This includes the *logic, limits,* and *parameters* in the rules, the *extent* of automation, and the *application* of the rules. The objective is to use all the *variables* (decisions) at our disposal to maximise the anticipated *net profit* which is the profit less the anticipated losses or claims associated with the approval process and the cost of the approval process itself.

4.4 Credit Approval

We now turn our attention to the specific problem in our case study, credit approval, create a model business process to support rule-based and human decision making. We conclude that the general problem, and that of our case study, consists of four sub-problems: information required, rule-building, interaction with human decision makers and operational adjustment. The first three are examples of off-line optimisation where we decide upon information, create, and optimise the rules and human interaction using historical or forecast data. An operational adjustment is a form of on-line optimisation where the rules are adjusted based on the situation.

There are four basic problems:

i. The optimal quantity of information to be collected, and in what order (the feature selection problem). With no information, we have grounds upon which to decide, so we can say – up to a point – that the information we have, the better decision can be made. However, if we have humans processing information, they need time to digest and assess, and this costs money. In addition, potential customers may find it intrusive and tiresome if we ask for too much and may decide to discontinue or abandon the transaction. This is a personal characteristic and not predictable on a case by case basis, so we should be mindful of the likelihood that this will happen, and that we will lose a potentially good customer.

ii. The CRA rules problem – finding that set of rules that direct the cases to the machine or human expert for the best outcome. The underlying classification problem for credit application has been widely researched, and there are tried and tested methods when we use the rules as the only means to decide. It gets interesting when we have the option to refer some instances to human decision makers. The rules have two functions; firstly, to decide on referral and secondly, to decide on the cases that are not referred. In this process, we determine the number of experts for any given caseload and case-mix.

iii. The dynamic interaction of machines (processing rules) and human experts. We may have identified the optimal information set for the anticipated case mix, but there will be cases where the optimal amount of information is case dependent. For example, a highly paid individual with a perfect credit rating should require less scrutiny (and therefore fewer attributes) before a positive decision is made than a less clear-cut applicant.

iv. The operational problem – finding methods to dynamically modify the rules when the caseload or case-mix changes. If the resultant change means that we need more experts, the processing rate will reduce, and the queue will increase. Not only will this reduce revenue in the short-term, in the medium-term customers may get tired of waiting. Conversely, if there is insufficient work for the experts, it would be better to give them additional cases if it is expected that they would get a better result than the rules.

4.4.1 Problem Definition

To maximise the financial benefit from a set of customer enquiries, the business rules that control the process must be designed so that the decisions are correct as far as possible, given the information available. Besides, the quantity and type of information gathered must be chosen carefully to strike the right balance between the value of the information and the cost of gathering and processing it.

The outcomes of the customer categorisation process are; (1) a set of customers accepted (2) a set of customers who withdraw from the process (3) a total net benefit from the customers served, and (4) an opportunity cost associated with the customers who are not served either because they dropped out, or because they were not selected. The process also has a cost (5) arising from the resources consumed in its component activities.

If the benefits (3) and the costs (4) and (5) can be quantified, the process is optimised when (3) – ((4) +(5)) has been maximised. In all cases, we need to classify the customer (or case) as accurately as possible while minimising the cost of classification, the opportunity cost of rejecting a customer that we should accept and the liability of accepting a customer we should not have accepted. There is also consideration around abandoned transactions; making the customer experience more onerous by asking for too much information can result in otherwise potentially valuable customer giving upon the process. Also, we must recognise that financial services providers have levels of engagement including on-line, customer service agents and experts, such as underwriters, each of which brings different costs and expertise to the problems.

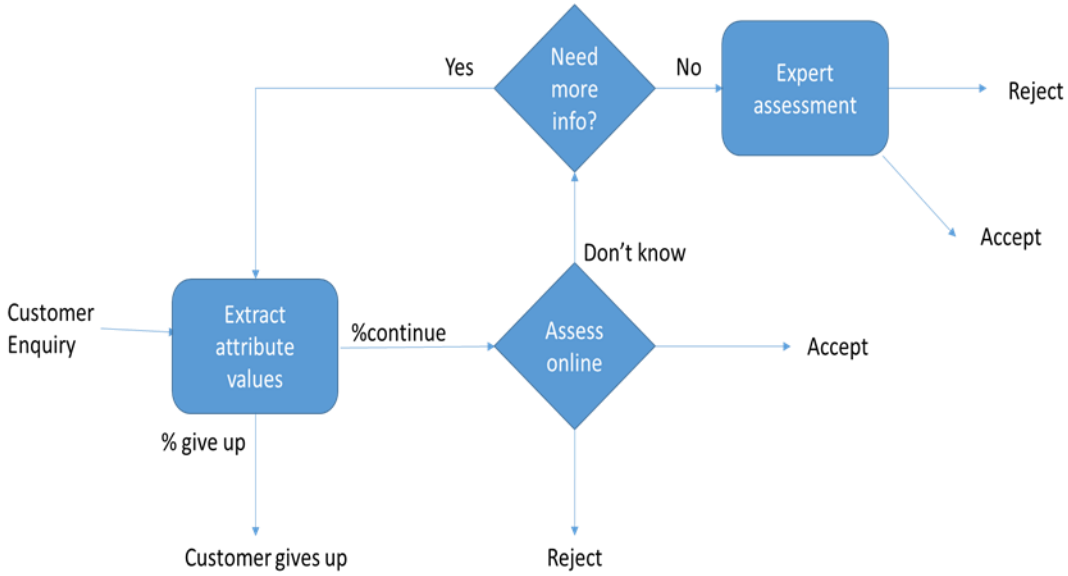

Figure 5 Business Process Flowchart

Figure 5 shows the business process, where the rounded squares are actions and the diamonds are decisions. The on-line assessment refers to automated business rules. As the customer enquiries (cases) progress through the workflow decisions are made based on the information with each case. There is an initial set of information and based on that information the rules decide whether to accept, reject, request further information, or refer to the expert. Also, a proportion of customers may give up as more and more information is requested.

Let $\{x_{i:}\ \epsilon\ I\}$ be the set of I cases. The complete information about a case is specified in terms of a fixed set of attributes $\{A_j: j\ \epsilon\ 1...m\}$. For a case x_p the value of its q^{th} attribute A_q is written a_{pq}.

The potential value g(i) of a case x_i is a function of its attribute values: $g(i) = g(a_{i1}, a_{i2}, \ldots, a_{im})$. If $g(i) > 0$ this is a good case, and if $g(i) < 0$, it's is a bad case. Good or bad is the classification of the case. The value of *g(i)* depends on two factors; firstly, whether the case is good or bad, and secondly, the potential profit (from a good case) or the potential loss (from a bad case).

We write *info*(i) to denote the set of attribute values currently known about case i. The process of extracting information about a case x_i (task "*Extract attribute values*") is formalised as an update to *info*(i). If the value for attribute A_j is extracted for case x_i we write:

$info(i) := info(i) \cup \{A_j = a_{ij}\}$

We can formalise the choice of which attributes to ask about as a function of the case, i, and the current information, info(i):

ask (i, *info*(i))

Thus, the result of the task *Extract attribute values* is formalised as:

$info(i) := info(i) \cup ask(i, info(i))$

If we denote by *exp* the current load and competence of the expert, then we can formalize the "*Assess online*" decision as a function

Assess (*exp, info*(i))

which returns one of the values {accept, reject, unknown}.

Similarly, the "*Need more info*" decision can be formalized as a function from *exp* and *info*(i)

need_info (*exp, info* (i))

which returns one of the results {yes, no}.

The outcomes of the task "Expert Assessment" cannot be optimised, but its performance can be monitored to set and reset, the value of *exp* as necessary. The cost of the experts is also an outcome of the task "Expert Assessment". Let us represent it as *exp_cost*.

If the set of customer enquiries is I, the set of customers who give up is U, the set of customer accepted is A and the set of customers rejected is R (so that $I = U \cup A \cup R$), the quality of the whole process is representable as the following expression:

$\text{sum}(g(\text{i}): \text{i} \in A) - \text{sum}(g(\text{i}): \text{i} \in U \cup R) - exp\_cost$

Naturally, when optimising the business rules defining "*assess*" and "*need more info*", as well as the behaviour of "*Extract attribute values*", the aim is to maximise the above objective function. In our formal setting, the optimisation is achieved by optimally specifying the functions "*assess*" and "*need_info*".

A similar problem that can be addressed in the same way is one of approvals, particularly for purchases or commencement of a contract, where different levels of staff have different approval levels. If a purchase/project is above a threshold level, it needs to go to a more senior member of staff.

In the case of loan approval, the customers are represented by $\{x_i\text{:i} \in I\}$ and the attributes $\{A_j\text{: j} \in 1..m\}$ are characteristics such as income, amount of loan, current debt level, credit score, home ownership status, etc. The outcomes g(i) are the expected net profit from accepting this customer. If the customer defaults, then $g(\text{i})<0$ and the loan is considered bad, if not, $g(\text{i}) > 0$ and the loan is good. The values of g are dependent on each case, but they can be estimated based on experience and historical data. For example, we can calculate the potential profit from a loan given the amount of loan and the interest rate. Similarly, we need the potential cost of a default, this time considering the amount, timing, interest rate and security.

The first decision we need to make is how much information we ask for at the initial application process, which gives us *ask* (i, *info*(i)) and *info*(i). We then consider the capability and capacity of the expert to determine the on-line assessment, which will be one of {accept, reject, unknown}. In the case of accept or reject the case is closed. In the case of unknown, the case is referred to the expert who may decide he needs more information based on the attributes of the case. In this process, we have a proportion of customers, U, who may give up during the process.

The process itself is capable of repetition whilst there is a net benefit of incurring more cost, either in the information gathering (where a proportion of – potentially good – customers give up) or in the information processing and decision making by the expert. Information that is known to be very helpful might be more onerous for the customer to supply and hence more likely to cause a customer to drop out. We do not have data that covers all of these elements. Still, the tendency for customers to drop out of on-line transactions is understood (Mägi, 2016), reproduced below, shows the tendency of customers to give up due to the time required to complete an on-line transaction, for example. Data specific to loan applications, and similar activities, is not published. We only have data regarding general on-line transactions and the impact of waiting time. All we can say is that the more we ask for, the longer it will take and make some limited deductions. A business would be able to collect further information relating the nature of the questions themselves to the rate of abandonment. Figure 6 shows the impact of waiting time on transaction abandonment based on a large number of different types of transaction.

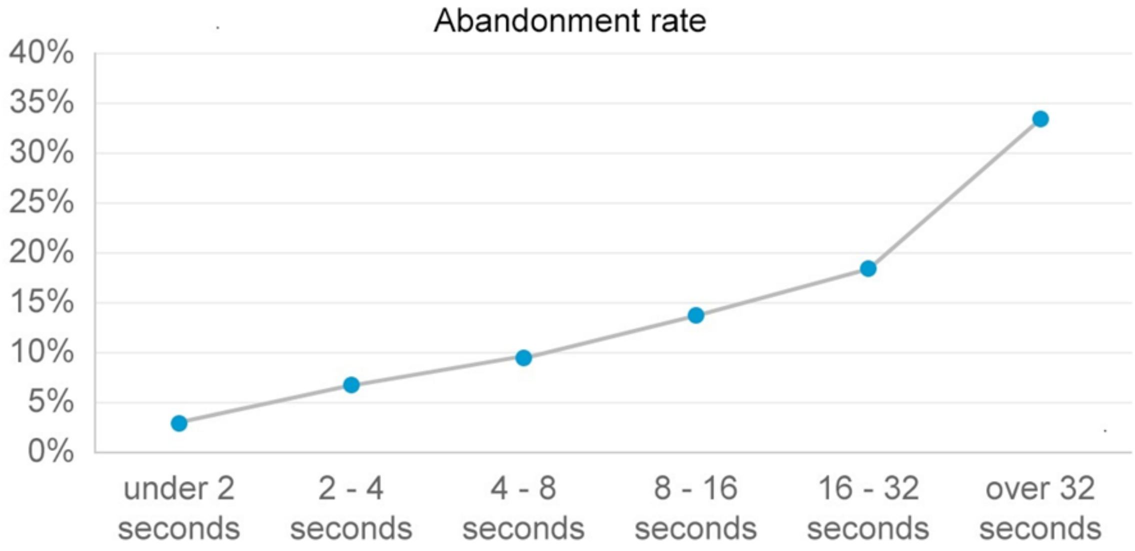

Figure 6 On-Line Customer Behaviour

The essential optimisation problem is to maximise expected benefit (good applications – those we wrongly reject – those that give up) against the expected cost (bad loans that we mistakenly accept + the cost of the expert).

4.4.2 Structure

The objective of the business process is to make decisions around cases. These decisions typically include accept (enter into a contract of some sort), reject (do not enter into a contract) or refer, which also identifies a destination resource, i.e. refer (destination).

In many cases, we have three types of resource:

- The machine, which follows the rules to the letter
- The agent, who has a limited amount of expertise and exercises a limited amount of judgement (for example a claims agent),
- The expert, who has the highest level of expertise and exercises complete judgement (for example, an underwriter)

There is no reason the expert level could not be further refined, as in the case of a major decision where a board of directors or even a shareholder vote may be required. But for this exercise, all potential actors are either agents or experts, and this structure is enough. Therefore, we need no more layers or resource categories, and this representation will generalise

However, as we shall see in Chapter 8, we only have data for one level of human interaction on which to test our theory. But if data were available, there would be value in including a lower cost human resource. To see this, we can extrapolate.

Assume that the machine is free, and the expert costs are positive. We have:

Expected net benefit of processing by machine (rules) = $Rgain$

Expected net benefit of processing by expert = $Egain$

Then is

$Egain - \ Rgain > Ecost$

We process by the expert

Else process by the rules

If $cost > 0$ we just need to find another human agent such that

$Again - Rgain > Acost$

For at least some of the cases.

4.4.3 Agent and Expert

The most straightforward process (without automation) is shown below. Customers (cases) are held in a queue and processed on a First In, First Out (FIFO) basis.

The agent processes every case and the cases divide into those that he must refer (based on a set of rules) to the expert and those that he can decide upon himself. Of these cases, he can accept, reject or refer. There is an apparent inefficiency here as cases that must be referred as handled twice, and each time a case is touched two sorts of costs are incurred: familiarisation and processing.

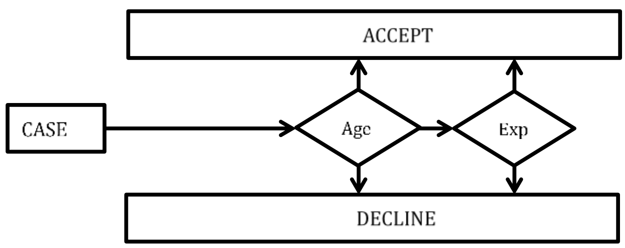

Figure 7 Basic Business Process

The next scenario is to add automatic (i.e. computerised) pre-processing to improve the performance of the organisation and reduce costs. The pre-processing is assumed to have a zero marginal cost to execute and can make decisions itself or direct cases to staff as appropriate.

Initial processing is done by a machine/computer that interprets any rule literally but at (effectively) zero cost and zero time. The rules that are applied lead to rejection, acceptance, or referral. The diagram is simplified as, in effect, it would be possible for:

- The computer refers a case directly to an expert (based on the rules)
- The agent can refer cases before they are entirely processed
- Cases could be referred to the agent by the expert

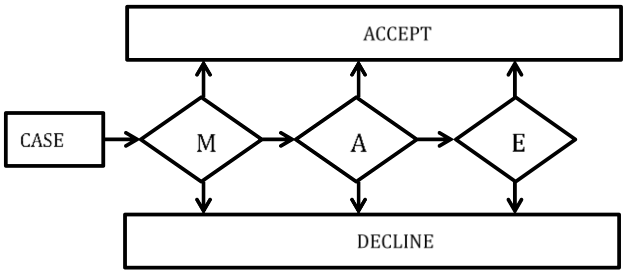

Figure 8 Machine, Agent and Expert

The triage process, shown below, can process cases in the queue.

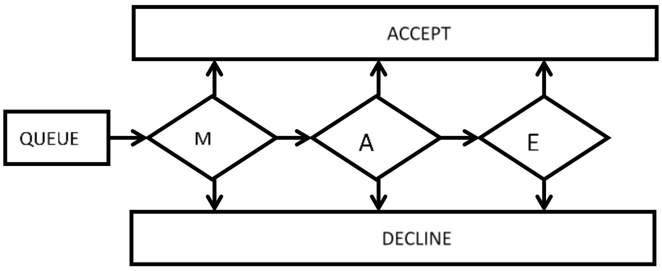

Figure 9 Augmented Business Process

This process takes advantage of the fact that computer processing is zero cost and enables the characteristics of a queue of cases (or customers) to be determined. In that case, it is possible to select cases from the queue in such a way we keep the agent and experts busy, reducing idle time and queue length. The computer processes the case immediately and at zero cost, and the rules can be adjusted such that the rate of referral precisely matches the expert resources available.

The objective here is to reduce the average waiting time and processing time, given a resource profile of agents and experts, with a proviso that no case should stay in the queue beyond a maximum time limit.

One of the questions is to what extent can we improve on the alternative situation where cases are sent immediately to sub-queues that wait for the attention of agents and experts.

4.4.4 Proposed Representation Approach

We can start with the assumption that the rules are applied to information supplied by the customer. For example, salary, age, etc. Real and integer values can be compared to maximum and minimum limits. Figure 10 below shows how a computer might be set up to do an initial screening process. The idea is to set limits wide enough such that only those cases where there is some doubt are referred for further consideration. The decision to reject or accept is made in the knowledge that it is correct, as far as possible.

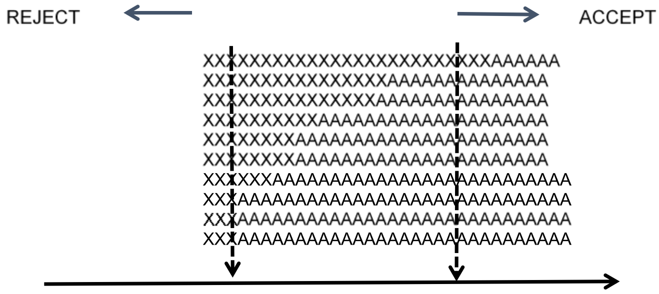

Figure 10 Initial Screening

The next stage is to refer cases, first to the agent. The agent is empowered to consider cases within specified limits. If a case exceeds a limit for one or more parameters, then the case must be referred to an expert. Note that these parameters could be binary (a claim in the last five years) and a positive response would mean that the Agent or expert would need to assess the case.

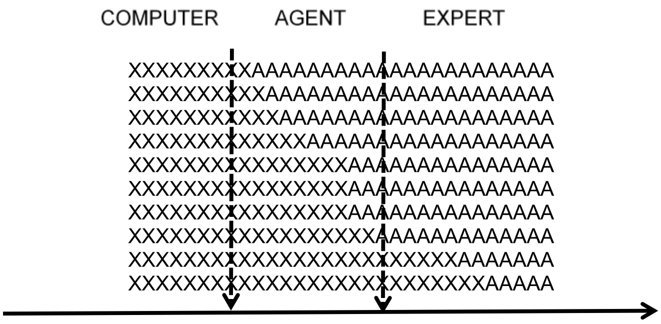

Figure 11 Subdivision of Limits

The problem above is essentially a classification problem. We have to derive limits such that we capture enough of the customers that we should accept without accepting too many that we should not be.

Thi process works at three levels:

i. At the machine level, the rules are simply interpreted

ii. At the agent level, the rules are relaxed such that some judgment (which is not necessarily infallible) can be brought to bear

iii. At the expert level, there are no rules, but judgement can be supposed to be more accurate, but still not necessarily infallible.

We need to represent:

- The rules
- The ability for agents and experts to make judgements

We first confine the problem to the following, generic, example:

- Customers are all different, and we required to maximise potential revenue while minimising cost and risk.
- Maximising revenue means that we offer the right level of service to each customer
- Minimising cost means we deal with customers efficiently and do not over-service them
- Minimising risk means we do not transact with customers where the costs, or expected costs, exceed the revenue

4.4.5 Attributes

The attributes are the information that is gathered for each case.

4.4.5.1 Real

Real attributes can take any number such as age, income or loan value.

4.4.5.2 Logical

Logical variables are either TRUE or FALSE. For example, if there has been a claim or credit default in the last five years. Logical attributes are a special case of nominal attributes.

4.4.5.3 Nominal

These include items like marital status (e.g., married, single, divorced), the highest level of education, employment status (employed or not), homeownership (renting, buying, own).

4.4.5.4 Ordinal

This is a class of nominal attributes where we have some order, because they are integers (number of children) or because some sort of order is implied.

It is common to treat nominal variables as ordinal, which can simplify the problem, especially where there is a positive correlation between the attribute and the outcome. For example, if we have education or housing status, we can logically order these as:

- High School (1), University Degree (2), Master's Degree (3), Doctorate (4) or
- Rent (1), Buy (2), Own (3)

4.4.5.5 Logical Operations

There are also tests that we need to carry out:

.EQ. Equal to

.LE. Less than or equal to

.GE. Greater than or equal to

4.4.5.6 Parameters

For logical variables, parameters are not required, but for Real and Integer we need limits. For example:

IF (Attribute. GE. Maximum Limit)

IF (Age. GE. 60)

For choice variables, we have to test against the allowable choices (usually a checkbox). For example:

IF (Attribute. EQ. Choice 3)

IF(Marital Status. EQ. Divorced)

4.4.5.7 Combination

These individual tests must be combined using logical Agents AND and OR to create rules that can be coded.

4.4.5.8 Representing Judgement

The machine has no judgement and implements the rules literally.

The agent and expert have judgement, and we need to represent that. The underlying assumptions are:

- The machine gives the agent a sample of cases,

- The expert receives cases from the machine and agent
- The role of the agent is to accept, reject or refer
- The agent decides, input by input, whether to accept, reject or refer

We need to create a representation of the judgement of the Agent and expert

For the agent, the task is to accept, reject or refer (to the expert). The task of the expert is to accept or reject. As the problem for the expert is more straightforward (no referral), we can concentrate on the agent.

There are two potential simple approaches:

The agent misclassifies a fixed proportion wrongly and refers a fixed percentage of cases: For example:

- 20% of cases are referred to the expert
- 5% of cases that should be accepted are classified as reject
- 5% of cases that should be rejected are classified as accept
- The remainder, 70% of cases are correctly classified

The actual percentages could be derived from historical data, and these would map well onto KPI's for correctness (getting the classification right) and efficiency (creating work for the expert). But given that one of the questions is how many cases to send to the agent, and how many to send directly to the expert, we may not assume that these percentages are fixed. The other problem with this approach is that we do not know which cases to get right. That is, if the correct proportion is 10% of cases, the next question is which 10%.

Another approach is to consider the attributes of a case as a whole. This is an example of a multi-attribute evaluation problem (Harte, 2001) and one way that we can model the decision is to create a linear combination of the attributes a_i, weights, w_i, and derive limits (this is very similar to the LENS model which is based on linear regression)

We choose limits, L_{reject} and L_{accept} , and weights, w_i, acting the attributes, a_i, such that:

$\sum a_i w_i < L_{reject}$ THEN reject

$L_{reject} < \sum a_i w_i < L_{accept}$ THEN refer

$\sum a_i w_i > L_{accept}$ THEN accept

The same model could be used for the expert, but with only the choice to reject or accept.

The weights themselves can be determined by comparison with actual decisions made by the agent over time. But they are not important at this stage, and we can assume that we assign sensible weights (or use scale the inputs) and adjust the threshold accordingly.

There are many other ways of modelling human judgement that could be considered later, but the valuable addition is to overlay judgement.

We must represent the fact that the expert has better judgement than the agent. We can use the same weights but different thresholds (on the basis that the expert has only two choices: accept and reject over a narrower range) or change the weights to get a more accurate answer.

There is, however, a problem with this approach. If we can accurately model the judgement of the agent and expert, why not just code that up in the machine and dispense with them? There are two reasons why not:

In a practical sense, information is often not of a form that can be processed by a machine. For example, for an insurance claim, we may ask about previous claims, and if the answer is yes, we invite the claimant to provide further information in free text. For insurance cover, we may ask about previous claims, convictions, etc. Similar considerations apply for a loan application.

There are factors that the agent and expert may consider that may not be presented on the claim form, such as the length of the relationship with the customer, previous dealings, or the expectation that a customer will complain.

For these reasons, a simple model of the agent and expert, where they get a percentage wrong, not only deals with these issues but these percentages could be determined by analysis of performance.

We can also investigate the impact of coding a weighted-sum criterion on the machine.

There are two other important considerations:

The first is the availability of information (that is, missing values). There is a relationship between the number of attributes and the quality of the decision. If no attributes are available the choice is uniformed and essentially random. As more attributes become available, this percentage can only increase. The question is one of determining the relationship between the available attributes and accuracy

This relationship could be derived from experience or data analysis. We can determine the prediction capability of each variable using the impurity criterion when that attribute, and only that attribute, is tested against its optimal limit.

For a real attribute a, we find that value of c such that maximises the purity function (based on the presence or absence of members that default) as follows:

$$\max(\Sigma p^2) = p_1^2 + p_2^2$$

$p_1 and\, p_2$ are the proportion of cases correctly classified in each of the sets defined by the classification criterion. The maximum value is when $p_1 = p_2 = 1$

$$\mathrm{p}_1 = \left(\frac{|\{default = true\}\, with\ a \le c|}{|total\ a \le c\ |}\right)$$

$$and\ p_2 = \left(\frac{|\{default = false\}\, with\ a > c|}{|total\ a > c\ |}\right)$$

Similarly, for a choice variable we have

$$\mathrm{p}_1 = \left(\frac{|\{default = true\}\, with\ a = choice1|}{|total\ a = choice1|}\right)$$

$$and\ \mathrm{p}_2 = \left(\frac{|\{default = false\}\, with\ a = choice2|}{|total\ a = choice2|}\right)$$

The next issue is how to implement this approach. There are two basic options:

- We ask the same information from every customer
- We have a basic set, and then ask for more depending on the information already collected

Note that ‘already collected’ could be in the context of the current transaction or based on profile information already available.

4.5 Summary

In this chapter, we have investigated some service business processes that are, or could be, governed by business rules. In such cases, the quality of decisions and subsequent outcomes are influenced by the nature of the business rules. Decisions can be binary (logical) or integer (choice), but for reasons of simplicity and availability of real data, we focus on binary decisions. We have also analysed the specific problem of credit approval from the perspective of business rule optimisation.

5 OPTIMISATION

5.1 Introduction

In this chapter, we define the optimisation problem in terms of the objective function, degrees of freedom and constraints. The objective function usually is the profit of the organisation, or, more correctly, expected profit as we are building rules in the anticipation that their application will maximise profit in the future. This profit function will be derived from past data. It will consist of probabilities and potential benefits (from accepting good cases), losses (from accepting bad cases), costs of processing information and the impact of abandoned transactions. The degrees of freedom are the choices we make regarding rules. These include the attributes we choose, the logical tests we apply, and any parameters used within the criteria (such as maximum or minimum limits). For CRA rules optimisation problem, there are no constraints. In the operational problem, we assume that the number of experts employed has been determined as a by-product of rule optimisation, and the objective is to adjust the rules to make the best use of them.

First, we define and formulate the business rule optimisation problem. We then determine the nature of the objective function, which will influence the choice of optimisation methods. Finally, we consider the choice of attributes and the interaction of the rules and human experts.

5.2 General Forms of Optimisation

As previously discussed, there are two types of optimisation (Ponstein, 2004). The most common form is on-line, or data driven where each situation is optimised individually. The other is off-line or static where we optimise rules, response or procedures that are then applied in every situation. The base case for BRO is primarily off-line; the objective is to find the optimal set of rules over any given historical or forecastable period. While we also investigate the ability of rules to adapt or adjust, we would argue that this is different to on-line optimisation –

where a new optimisation problem is solved for each situation - and should not be confused with it.

5.2.1 Off-line or Static

The off-line (static) optimisation problem is to create an invariant function that acts on the inputs to control a process to create an optimum set of outcomes over a given set of scenarios. There is no suggestion that each situation will be optimised, per se, but that overall, the set of outcomes will be optimal. An example of static optimisation is determining the allocation of tasks to resources based on rules on location (of the task) and geographical areas (assigned to resources). For example, see Maciejewski (2014):

Maximise the expected value of $f(y, x, u){:}\, c(x, u)\ and\ y = r(u)$ W.R.T. r

Where:

f = the economic or financial outcome of interest. For example, profit or revenue

u = input variables that are given for any situation. For example, customer characteristics or weather

y = control or decision variables. For example, choice of customer, price of goods. Essentially, the response to the situation

x = dependent (outcome) variables. For example, quality and service levels

c = customer service and compliance (with statutory instruments and prescribed business rules) constraints.

r = invariant (CRA) business rules function

The CRA rules consist of a set of parameters, operators and logic and the choice of these is the off-line optimisation problem. This is a potentially large optimisation problem, and a practical approach to its solution is:

- Start from the data (knowledge base) and rules and calculations that the business currently uses
- Identify the operators and parameters in the rules and calculations that could be optimised or added
- Determine the objective function and express that as a function of the parameters and choices (above)

- Formulate and solve the optimisation problem over a set of the historical or expected set of inputs

  In the case study, for example, we already had the baseline from the accepted cases and outcomes, the additional rules were around the allocation of cases to the experts, and the parameters were the threshold values in those rules.

5.2.2 On-line or Data Driven

This form of optimisation is more common, and it is useful to make the comparison here. For each situation (set of inputs) algorithm creates the optimal control or decision that maximises the value of the single outcome. While in 5.2.1, there would be (invariant) rules around the allocation of jobs and resources based on geography, the situational optimum would allocate jobs to resources based purely on factors such as capability and efficiency. For example, see Bubeck (2011) and Jaillet et al. (2010).). The general form of an on-line optimisation where we optimise each situation independently is given by:

Maximise $f(y, x, u)$: $c(x, u) = 0$ W.R.T. y

With the same variable definitions as the previous section. Online optimisation has some advantages as each situation can be optimised. However, there are disadvantages, such as:

- The requirement for most – if not all – of the input data to be correct
- It can difficult to understand how the answer was arrived at (mainly when constraints interact)
- There may not be enough time to complete the necessary calculations

The significant difference between the two forms of optimisation is that the control variables, y, are calculated directly from the inputs, u, using the business rules. This calculation is straightforward, faster, and more reliable. And, if the rules are constructed using methods such as weighted sums or logical tests, then the results will be easy to understand.

5.3 Definition

For simplicity and clarity, the discussion below applies to the process of loan or credit application, but the basic structure applies to similar problems identified in

Chapter 5. It is also worth pointing out that even though the final outcome may be binary, this is not just a classification problem as we have to decide, for each case, whether to refer and this depends on the expected quality of the decision of the rules, the performance of the expert, and the cost of the expert. We also have the option to curtail the process and decide with less than the full data set, should the expected costs outweigh the expected benefits.

Let $\{x_{i:} \ \epsilon \ I\}$ be the set of I cases. The complete information about a case is specified in terms of a fixed set of attributes $\{A_j: j \ \epsilon \ 1...m\}$. For a case x_p the value of its q^{th} attribute A_q is written a_{pq} , and the set of attributes for case p is written A_p.

The potential value g(i) of a case x_i is a function of its attribute values: g(i) = g (a_{i1}, a_{i2}, …,a_{im}). If g(i) > 0 this is a good case, and if g(i) <0, it is a bad case. Good or bad is the classification of the case. The value of *g(i)* depends on two factors; firstly, whether the case is good or bad, and secondly, the potential profit (from a good case) or the potential loss (from a bad case).

We also have two types of determination for x_i. The determination made by the rules:

$rd_i \ \in \ RD = \{ACCEPT, REJECT, REFER\}$.

and the determination made by the human:

$hd_i \ \in \ \ HD = \{NONE, ACCEPT, REJECT\}$

We represent the business rule by a function $rd: A \rightarrow \ RD$ with $rd(i) = rd(A_i)$ and similarly, we represent the human decision making by a function $hd: A \rightarrow HD$ with $hd(i) = hd(A_i)$

We also have:

exp_cost = the expense of the human processing the case

We can this determine the expected economic value of a case and its determination denoted by $p(i) = p(g(i), rd(i), hd(i), exp\_cost)$.

In conclusion, we are seeking to find that set of rules that maximises the objective, P, for the n cases in the data set of interest with respect to the rules function:

$$P_{max} = \max_{rd_i} \sum_I p(i)$$

For example, suppose one of the rules is the minimum income level at which we accept an applicant for a loan, I_{LOWER}. We then have a series of values for G as a function of I_{LOWER}. When $I_{LOWER} = 0$ we accept all applicants and the value of G is the outcomes we would expect if we accept every case. As I_{LOWER} increases, we would expect the value of G to increase as we remove applicants on a low income that have a higher proportion of bad outcomes in the data set. With a very high value of I_{LOWER} we have fewer and fewer applicants approved, and G starts to decrease. As such, there is a value for I_{LOWER} at which G is maximised.

5.4 Problem Formulation

To formulate this as an optimisation problem we need to determine what we are maximising (such as the financial contribution to the organisation) and how we are going to do it (choosing the information we request and the parameters of the business rules). The former is the objective function and the latter are the degrees of freedom.

5.4.1 Degrees of Freedom

In the context of the problem defined above, we have two essential sets of choices:

i. The optimal attributes we require (the elements of A in 5.3)

ii. The optimal business rules (the function rd in 5.3)

Typically, some attributes are already known to make a difference to the outcome. This knowledge enables us to make decisions, and the issue is one of attribute selection.

We could choose any function for the rules, but in standard business practice allocation and reasoning business rules are expressed as a chain of IF-THEN-ELSE logic (like a decision tree) with logical tests based on the numerical value or category of the attributes (Gottesdiener, 1997). In some cases, there is value in including functions of attributes or derived attributes as additional inputs to the rules; these are the computation rules. Examples include the ratio of loan: income in the case of mortgage applications, or the number of standard deviations from the mean, in case of quality control.

5.4.2 Objective Function

Earlier, we identified that the objective could be multicriteria with profit and customer satisfaction as typical quantities of interest. Such multicriteria problems are typically addressed by a weighted sum to create one overall objective function.

If we take this approach, we are looking to maximise the expected value of the objective function which can be expressed as the expected values of:

Gains from good cases accepted

$\sum_{g(i)>0 \ and \ (rd_i \ OR \ hd_i)=ACCEPT} g_i$

- losses from bad case accepted

$\sum_{\mathrm{g(i)}<0 \ and \ rd_i \ OR \ hd_i=ACCEPT} \ g(i)$

- cost of processing information

$\sum_{hd_i=ACCEPT \ OR \ REJECT} exp\_cost$

- abandoned transactions

$at(m) \sum_{g(i)>0c} \quad g(i)$

Where (for each case i) that is accepted we have:

g_i = potential profit or loss

exp_cost = cost of expert assessment

hd_i = expert decision

rd_i = rules decision

$at(m)$ is a function that estimates the probability that a customer will abandon the transaction as the process requests more information. This function can only be an approximation because we do not know which customers or whether we would have accepted them or not, but for economic and customer satisfaction reasons, we need to be mindful of the number of attributes we request.

In the general case, the objective function is to maximise the expected value of profit. Using the training set we optimise the value across the training set.

The nature of the function determines the optimisation techniques that can be applied. For example, to use gradient methods (linear programming or non-linear programming), we require it to be continuously differentiable. And there are certain functions, such as linear and quadratic, that amenable to more specialised methods that can be easier to solve that others.

We have a variable, x, such as income (in the case of a loan) or size of claim (in the case of an insurance claim) that has an impact on the probability of a good or bad outcome.

For any interval (a_1, a_2) or category, a, we need to determine the probability of default and the attribute, x, belongs to a:

$$d(a) = p(default\ and\ x \in a) = p(default|x \in a)\ p(x \in a)$$

The explanation of this is: the probability of default in any category or interval, a, is the proportion of defaults in the population as a whole, multiplied by the proportion of default cases that are in the interval divided by the proportion of all cases in the interval. This calculation also gives us an efficient way of determining these probabilities from a training set or past data. Table 8 shows the probabilities of default for each of the eight categories.

Category	1	2	3	4	5	6	7	8	Total
Default	0	1	2	5	8	10	12	12	50
Not default	15	14	8	6	3	2	1	1	50
$p(x \in a)$	0.15	0.15	0.10	0.11	0.11	0.12	0.13	0.13	1
$p(x \in a\|default)$	0	.02	.04	.10	.16	.20	.24	.24	1
$p(default\|x \in a)$	0.00	.067	.20	.46	.73	.83	.92	.92	N/A

Table 8 Conditional Probabilities of Default

The probability of default in any one category is the proportion of default cases. The important part is that we can use the formula when we are dealing with intervals, rather than categories.

We have

$$f(a) = p(default\ in\ a) = p(default|x \in a)p(x \in a)$$

There are alternative ways to look at the problem:

- We have a population, and for each attribute, good and bad outcomes each have a known probability. For n attributes, we will have n+1 distributions: the base variable and probability of default for each of n attributes

- We have two populations, representing good and bad outcomes, and we have n probability distributions concerning the n attributes

In the first case we have:

$$prob(default) a \leq x \leq b = \int_a^b f(u)du$$

Where $f(u)$ is the probability density function as above.

We also have profit and loss as functions of x:

$profit = p$ (from cases that do not default)

$loss = l$ (from cases that default)

$default\ d(x) = 1\ for\ default\ or\ d(x) = 0\ otherwise$

$np(x) = (1 - d(x)) *\ p(x) -\ d(x) * l(x)$

$netprofit = np$ (the net result of profits and losses)

In the second case we have:

$$prob(default\ and\ a \leq x \leq b) = \int_a^b h(u)du$$

Where $h(u)$ is the probability density function for bad outcomes, and:

$$prob(non\ default\ and\ a \leq x \leq b) = \int_a^b g(u)du$$

Where $g(u)$ is the probability density function for good outcomes.

We will use this second representation in the rest of this chapter.

5.4.3 Nature of the Objective Function

The nature of the objective function has an impact on our choice of optimisation methods. For example, if the function is differentiable, we can use gradient based methods; if not, we are restricted to methods that only use function values, and so on.

Let us consider a simple rule on a single attribute. Suppose we have a rule based on a single value c, such that

$If\ (x \leq c)$

we accept the customer (and any potential profit and loss) then

$np(x) = g(x)p(x) - h(x)l(x)$

And if

$If(x \geq c)$

we reject the customer, then

$np(x) = 0$

The profit function is given by the expected value of the gains (from good customers that we accept) minus the losses (from bad customers we accept):

$$Exp((np)(c)) = \int_{-\infty}^{c} g(u)\, p(u) - h(u)l(u)du$$

So $Exp(np)(c))$ is the expected value of net profit as a function of the cutoff, c.

We can differentiate w.r.t c:

$\frac{d}{dc} Exp(np)(c) = g(c)p(c) - h(c)l(c)$

And we have a stationary point when

$g(c)p(c) - h(c)l(c) = 0$

And this will be a maximum/minimum when

$$\frac{d}{dc}\big(g(c)p(c) - h(c)l(c)\big) < 0 /> 0$$

For example, if g and h follow the normal distribution:

$$f(x) = \frac{1}{\sigma\sqrt{2}} \exp \frac{-1}{2}\left[\frac{x-\mu}{\sigma}\right]^2$$

If we set the standard deviation of both to unity and have the mean of g equal to zero and the mean of h equal to unity, we would expect a good separation at a point between 0 and 1. Let us also assume that p and l are equal to unity.

Using the formula above we have a stationary point at c where:

$$\frac{1}{\sqrt{2}} \exp \frac{-1}{2} c^2 - \frac{1}{\sqrt{2}} \exp \frac{-1}{2} (c-1)^2 = 0 \Rightarrow$$

$c^2 = (c-1)^2 \Rightarrow c = 1/2$

This point is a maximum that implies ½ should be the maximum cut-off point.

For the multi-dimensional case (where we have n attributes) we can apply the same logic, with h(x) being the joint probability density function:

$$prob(default\ and\ a_1 \leq x_1 \leq b_1, \ldots, a_n \leq x_n \leq b_n) = \int_{a_1}^{b_1} \ldots \int_{a_n}^{b_n} h(u)du$$

$$Exp((np)(c)) = \int_{-\infty}^{c_1} .. \int_{-\infty}^{c_n} g(u)p(u) - h(u)l(u)\partial u_1 .. \partial u_n$$

And in this case, we have:

$$\nabla(Exp(np)(c))_i = \int_{-\infty}^{c_j} .. \int_{-\infty}^{c_k} (g(c_i, \bar{u})p(c_i, \bar{u}) - h(c_i, \bar{u})l(c_i, \bar{u}))\partial u_j .. \partial u_k$$

Where $0 \leq j < k \leq n\ and\ j \neq i, k \neq i, \bar{u} = (u_1, \ldots u_n) - u_i$

The function has a stationary point when $\nabla(Exp(np)(c)) = \mathbf{0}$ and this is a maximum/minimum when $\nabla^2(Exp(np)(c))$ is negative semi-definite/positive semi-definite.

In the case above, we are looking at the vector, c, that divides the n-dimensional space to maximise the profit function. This calculation applies to the rule that states that all the cases must satisfy the condition (the AND rule).

We express more complex rules, such as combinations of AND and OR rules in a similar form. For example, we can convert OR rules into a series of AND rules and so on. Similarly, we can have different cut off values depending on which side of each cut-off the attributes lie. The resulting set of rules are like those of a decision tree as we shall see in Chapter 7.

This analysis is very much a theoretical exercise because determining a joint probability density function is very difficult without making assumptions around independence. If we assume independence and the same potential profit and loss from each case, we have:

$$prob(default\ and\ a_1 \leq x_1 \leq b_1, \ldots, a_n \leq x_n \leq b_n) = \prod_{l=1}^{n} \int_{a_l}^{b_l} h_l(u)du$$

And so

$$Exp((np)(c)) = p\prod_{l=1}^{n} \int_{-\infty}^{c_l} g_l(u)du - l\prod_{l=1}^{n} \int_{-\infty}^{c_l} h_l(u)du$$

And

$$\nabla(Exp(np)(c))_i = pg_i(c_i)\prod_{l=j}^{k}\int_{-\infty}^{c_l} g_l(u_l)du - lh_i(c_i)\prod_{l=j}^{k}\int_{-\infty}^{c_l} h_l(u_l)du$$

Where $0 \leq j < k \leq n \; and \; j \neq i, k \neq i$

And the same considerations apply for stationary points, and maxima and minima. To apply this approach, we only need the probability distribution of each attribute which is a much more realistic prospect when we have a large data set.

However, we can conclude that the objective function is differentiable. Also, the derivative is the combination of probability density functions (which are normally continuous and, at worst, right continuous), and cost functions. So, if the cost functions are continuous, we have a differentiable objective function, and the derivative is right continuous. This means that efficient, gradient-based, optimisation methods may be suitable, but we cannot be certain.

5.4.4 Data Requirements

To create optimal business rules, we need a data set of cases, determinations, and outcomes. It is important to note that we are not just interested in determinations as these will not necessarily be accurate, as humans make mistakes. Data on outcomes may be harder to find. For example, borrowers may default later; diseases may be misdiagnosed and eventuate later, children can be returned to their parents and subsequently abused. Also, if we are considering the place of human experts in any optimised system, we need to know how well they perform.

5.4.5 Choice of Attributes

The classical feature selection problem considers the performance of the rules with different attributes. However, in our case, we are not incurring costs for every attribute we might have at our disposal, we are incurring costs for the attributes that we ask for and use. At some point, the expected benefit of asking another question will be exceeded by the costs of information processing and the probability that the transaction will be abandoned.

So, if we ask for attributes as the rules need them, there is no feature selection problem in the classical sense; we simply stop asking for more information when its value is outweighed by the associated costs.

5.4.6 Incorporating Human Experts

One of the potential decisions of the business rules is to refer the case to a human expert. In this case, we need to know whether it is worth incurring the additional cost of the human expert. The LENS model provides a framework to model and understand the accuracy of the rules and the accuracy of the human expert (true positives, false positives, etc.). Essentially, we need to know the probability that a case is good and the probability that a case is good if it is deemed to be by a human expert, and the expected benefits and costs for each.

As originally developed, the LENS model considers the (judgements of) the human decision maker and reality. It uses linear regression to model judgements and actual and computes the overall (average) accuracy of the models and the judgements. To decide on referral, we need to extend the model to calculate the expected accuracy of the human on a case-by-case basis and add the rules as an additional method of determining a judgement. We also need to calculate the expected accuracy of the rules. Given that the outcome of this exercise is ultimately Accept or Reject (binary) we substitute logistic regression for linear regression and add an extra dimension in the form of the rules acting on the attributes to produce an additional judgement. We also utilise the ability of logistic regression to calculate the probability that (a) the expert (or the rules) determine that a case should be accepted and (b) the probability that this judgement is correct or otherwise.

Note that it is important to compare the cases where the rules accept and reject as in either case a human decision may be cost-effective, for example, if the rules cannot effectively decide.

5.5 Summary

In this chapter we have defined the optimisation problem in terms of objective function, degrees of freedom and constraint. We have also introduced the concept of business rules working with human decision makers and explored the nature of the functions that make up the objective. We conclude that the objective function is continuous and differentiable. However, the derivative is only right continuous which means that optimisation methods that rely on the objective being continuously differentiable may not be effective.

6 INFORMATION

6.1 Introduction

In this chapter, we examine the issue of information in terms of attribute selection and the impact of the amount of information we request on the potential customers. There is a real optimisation problem here as:

- The more information we have, up to a point, the more accurate can the assessment be, and thus potential profits will be higher
- The more information we ask for, the more likely it is that a potential customer will abandon the transaction

W then analyse data on transaction abandonment and show that there is a good fit with the Weibull distribution, which is often used to model survival processes.

Finally, we address the problem of missing data; whilst we may have data on expert judgements and outcomes, we will not have data on outcomes for cases that are rejected.

6.2 Attribute Selection

An important question when business rules use information to make decisions is the amount of information that is needed/requested.

Clearly, with no information we are unable to decide (apart from an arbitrary one) and thus add value. For example, if we have several people who want a bank loan, having no information would not enable us to choose a subset that is any better than all the applicants, or any arbitrary subset.

At the other end of the scale, asking for a large amount of information has disadvantages:

- There are costs associated with gathering information, for example interviewing face-to-face or following up when errors are made on forms
- There are costs associated with processing information, for example information needs to be read and assimilated for any decisions that are made by humans

- Asking for too much information may put customers off and/or they may abandon a transaction. This is particularly true in life insurance (some policies do not require a medical) or mortgages (some lenders require less documentation that others – so-called low doc mortgages) where asking for less information is considered a competitive advantage and presented as a benefit to the customer.

  We can express this as:

  Profit = value of good outcomes – cost of bad outcomes – loss of good customers - processing costs – net value of abandoned transactions

  And

  Expected profit = expected value of the RHS above

  Where:

- Value of good outcomes = profit from customers where we make a good decision (that is, we judge a good customer as good and accept them)
- Cost of bad outcomes = loss from customers where we make a bad decision (that is, we judge a bad customer as good and accept them)
- Loss of good customers = lost profit from good customers that we reject in error (that is, we judge a good customer as bad and reject them)
- Processing costs = costs we (the business) incur from evaluating information
- Abandoned transactions = net profit that we lose when customers do not proceed. Note that abandoned transactions can include customers that would have turned out to be bad, so we consider the net profit

There may also be customers who are put off and do not even start the process. We choose to ignore this on the following grounds:

- We can measure abandoned transactions; we do not know who did not even start
- In an on-line application, it is often not obvious how long the process is before you start.
- With a paper form, the complexity and difficulty can sometimes be determined, and the process abandoned without the business finding out (and/or recording the event systematically), but as paper forms are in terminal decline, this will become less significant

This creates some interesting issues:

- The (marginal) cost of processing information on a computer is negligible (UK Cabinet Office, 2012), so once we decide what information to ask for, the computer may as well process it all
- The main task of the computer is to decide whether to accept, reject or refer a case. On referral the expert or agent can review all the information, at which point we incur their costs which, we may assume, is related to the amount of information. We could, of course, make this more sophisticated and consider an initial cost, independent of the amount of information to reflect familiarisation, and then have individual additional costs arrived at via data analysis. In addition, the accuracy or their decision may be related to the amount and type of information available. It follows that processing costs should only be considered for those cases that require a determination of the agent or expert
- The most we know about potential customers who abandon transactions is the information that they have input up the point they cease. The likelihood that they do may be a function of the type of customer (which may be revealed by the information gathered up to the point of abandonment depending on the order of the questions), the amount of information required, and/or the perceived difficulty or time taken to gather and input it. If we had a more sophisticated understanding about customers and their propensity to abandon, there is potential to treat them differently. For example, we may refer early (as we gather more information and make the assessment) and engage directly in the anticipation that this will reduce the likelihood that the customer does not proceed. We do not have the required data, but it may be an interesting area for further study.
- Let us assume for simplicity that (a) there is no material difference between customers who abandon and those that do not and (b) that the likelihood of an abandoned transaction is a function of the amount of information required/requested.

In many customer interactions, the business requires customers to provide information. Transactions such as approval for credit, insurance claims, and entering into lease agreements are typical examples. Similar considerations apply to medical diagnosis and selecting people for employment. In these situations, there will be a best set of information and we need to find a way to determine that.

So, let us define best as that subset that maximises the expected net benefits to the organisations that is dealing with the customer. Let us suppose that we have identified n possible pieces of information that we could ask for. Denote this by I_n.

In the case of credit checking we may have:

I_1 = income

I_2 = loan value

I_3 = credit score

Etc

Now suppose that we can identify:

- The potential value of good customers
- The expected loss from the bad customers
- The costs of processing the information that we have collected
- The impact of customers deciding not to proceed, or abandoning the process

At the highest level we have:

$$
\begin{aligned}
&Profit, p(I_i)\\
&= \ Value\ of\ good\ customers\ that\ proceed\ and\ that\ we\ select\\
&- \ Loss\ from\ bad\ customers\ that\ proceed\ and\ that\ we\ select\\
&- \ Processing\ cost\ for\ all\ customers\ that\ proceed\\
&- Net\ loss\ from\ customers\ who\ abandon
\end{aligned}
$$

$Where\ I_i \ \subset \ I_n\ and\ is\ the\ set\ of\ chosen\ variables$

What we are looking for is the subset I_i of all potential information inputs I_n that maximises p.

For example, if we have 100 customers, (50 good and 50 bad) and we ask for 5 pieces of information we might have:

- 20% of customers decide not to proceed (assume good and bad are in the same proportion as the overall population)
- We accept 37 out of the potential 40 good customers, that give us a profit of $500 each

- We also accept 4 bad customers out of a total of 40 that will create a loss of $200 each
- We incur processing costs of $25 per customer ($5 per piece of information)

In this case, total net profit is equal to:

$$\$(37 * 500 - 4 * 200 - 80 * 25) = \$16{,}500$$

Note that a simplifying assumption has been made around transaction abandonment. Without further information we have assumed that customers who do not proceed reflect the overall population (good and bad in the subset are in the same proportion as the population). If this is the case, a simple reduction on the net return calculation is all that is required. The data required to do any better is purely hypothetical; we do not have an outcome for customers that do not proceed. One potential method is to use a logistic regression model to calculate the probability that the customer would have been good (or bad) on the limited data (reduced set of attributes) that they have input in the process up to point they decide not to proceed. This method of predicting probabilities using logistic regression on a reduced set of attributes is derived and discussed further in Chapter 11. We do not have any data on the that we can find – it probably exists in proprietary data sets only – but not publicly available.

6.2.1 Solution Methods

There are a several ways to determine the optimal set of features, but unfortunately there is not any research that includes transaction abandonment. The choices are:

- Wrapper methods (Hall et al, 2013) where we select attributes on their merit as determined by the overall objective function that we are interested in. In effect, we select the attributes and carry out a complete optimisation to determine the profit or benefit, considering all the elements
- A filter method (Hall et al, 2013), where we use other metrics, such as information gain, to select the best subset of attributes

In our situation, for reasons of practicality, we will opt, if possible, for a solution which is reliable and not compute intensive. In addition, it is difficult to see how we could factor in the impact of transaction abandonment in the filter method. As such, the wrapper method is preferable, at least as first choice.

The other issue is the algorithm itself.

- Forward selection (Ververidis, 2005) consists of evaluating the profit function for each attribute, choosing the best, and then repeating the process until the profit function stops increasing, or increases by only a small amount.
- Backward elimination (Koller et al, 1996) is the opposite; staring with all the attributes and eliminating the least significant first until such time as the profit function starts to decrease.

The advantage of forward selection is efficiency and speed. The method is efficient as it does not need to evaluate the objective function after the process has terminated, often without including all the attributes. And with the wrapper method it is much faster as it is solving a much smaller optimisation problem, initially with one optimisation variable, then two, then three and so on. The disadvantage is that it may prematurely stop as the incremental value of the additional attributes, on their own, may be small or zero BUT when taken together the incremental value of two or more may not be. Figure 12 shows an example: when we split on two (real) attributes, 1 or 2, individually we get poor separation. Splitting on both gives good separation and information.

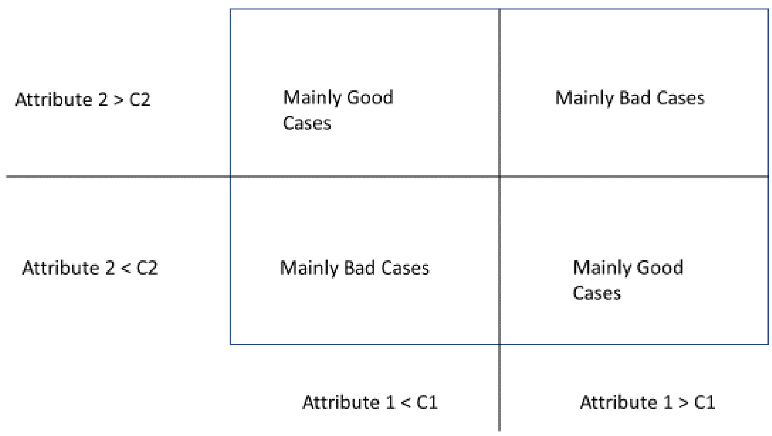

Figure 12 Problem for Forward Selection

Backward elimination avoids this problem as, initially, all the attributes are included and then eliminated one-by-one. The disadvantage, at least with the wrapper method, is that the initial optimisation problem has all the attributes as optimisation variables and is much more compute intensive.

Correlation-based feature selection (CFS) (Hall, 1999 and Hall, 2000) has been developed to address this problem. It identifies attributes (referred to as features in machine learning) that are strongly correlated to the classification, and the least correlated to the others. The results in a merit function that is used to create subsets, which are then selected using forward selection. Effectively we are grouping those attributes where the whole is greater than the sum of the parts.

- CFS uses a merit function to measure each subset, defined by:
- $Merit = k\widehat{r_{cf}}/\sqrt{k + k(k-1)\widehat{r_{ff}}}$
- Where:
- k is the number of attributes
- $\widehat{r_{cf}}$ is the average class: feature correlation and
- r_{cf} is the average feature: feature correlation

6.3 Transaction Abandonment

According to Technopedia (2011):

> Abandonment is an e-commerce term that refers to a situation when a visitor accesses a website but terminates any actions by leaving the page. The abandoned activities the Web host desires may include purchasing a product or **service** or completing an online survey.

Whilst it as an e-commerce term, it is reasonable to apply the same idea to any form of interaction where data is requested from a potential customer. However, until now it has not been possible to get large amounts of objective data on the phenomenon. Whilst people could be asked about their behaviour when using other means, such as paper forms, it is only now that we can measure every customer interaction in the on-line situation.

Whilst abandoning a purchase may be more likely than abandoning a loan application, the amount of information involved in deciding a loan application is much greater than that for a purchase, and indeed customers are happy to give more information. However having scaled up the quantities of information, it is our best assumption that the data about the rate of abandonment scales up for loan applications as well – there is no other openly available data that contradicts this assumption, as far as we know.

There is a lot of analysis on why on-line transactions are abandoned, as this is a major cause of concern for on-line retailers and has been addressed a marketing or social behaviour problem. For example, see Rajamma et al (2009), Van de Geest (2016), and Xu et al (2015). However, transaction abandonment has not been addressed specifically as a problem in human computer interaction which is rather surprising. This could be because there is little publicly available data about the relationship between time and abandonment.

In the research that has been done, time taken is typically the third most important reason for a transaction to be abandoned. This is shown in Figure 13.

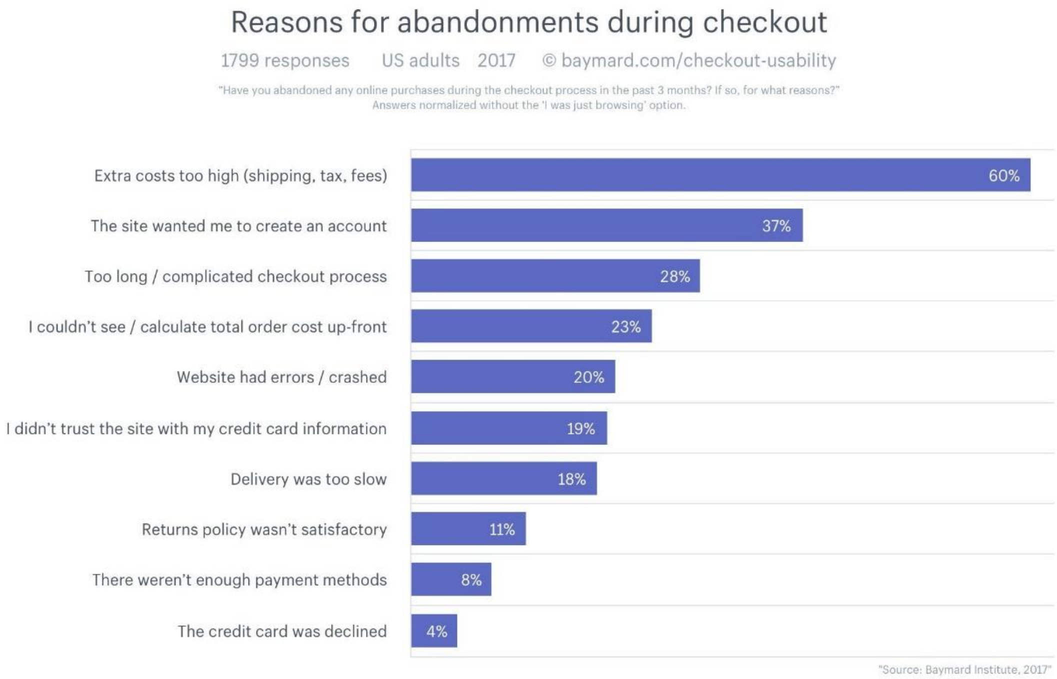

Figure 13 Reasons for Abandonment

Other research has been carried out in related areas, such as video viewing (Adage, 2019):

> The research yielded some compelling findings, including surprisingly high levels of initial viewer abandonment. For instance, our sample showed that, on average, nearly 20% of the audience that starts watching a given video clip will abandon it within the first 10 seconds of playback.

So, if your online video campaign has 10 million viewers, 2 million of them saw less than 10 seconds of it.

In general, **viewer abandonment appears to be a function of time spent in-stream and follows a relatively predictable trajectory**.

Below is a graph that summarizes average video abandonment by time spent viewing, which shows a consistent rate of viewer drop-off. Within the first 30 seconds of a video, you can expect to lose 33% of your viewers. At 60 seconds, 44% of the audience that started viewing the clip will have left. And so on.

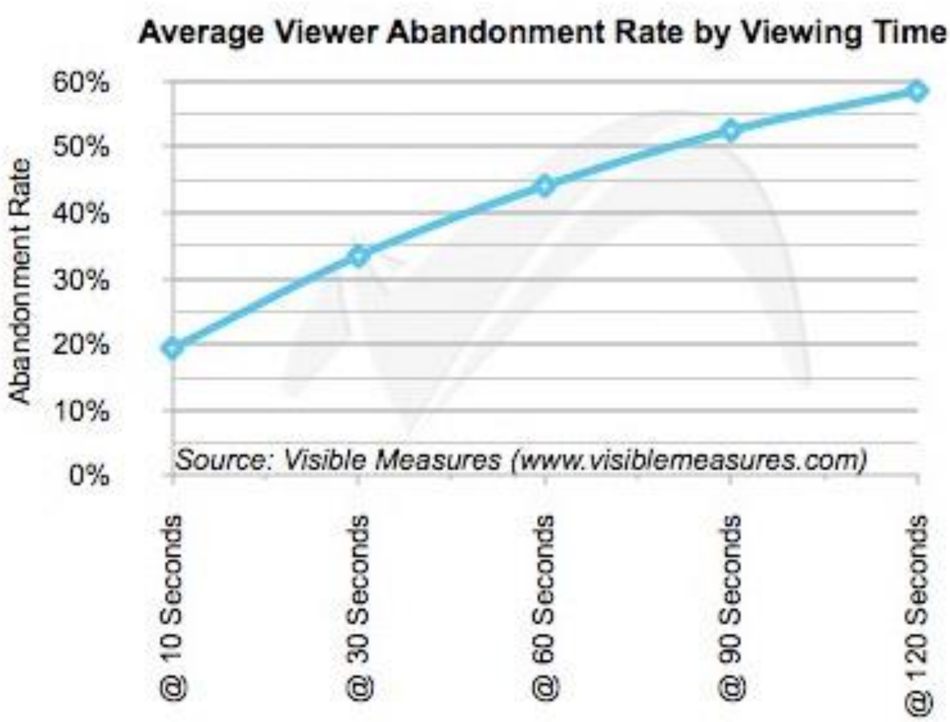

Figure 14 Graph of Viewing Abandonment v Time

The research by Mägi (2016) is more general and looks at 500 websites with some 5 Billion interactions. Whenever a user aborted an operation, it triggered a specific event captured by Plumbr, a software product that captures web interactions. Based on these events, the author aggregated, cleaned and clustered the data.

To interpret the data visualized in the chart above, let us look at the following examples:

- With response times less than 2 seconds, 3% of the users abandoned such operations.
- With operations taking from 2 up to 4 seconds, the number of users aborting the transactions grew to 6%.
- When the user-initiated operation took more than 32 seconds to complete, 34% of the users just gave up.

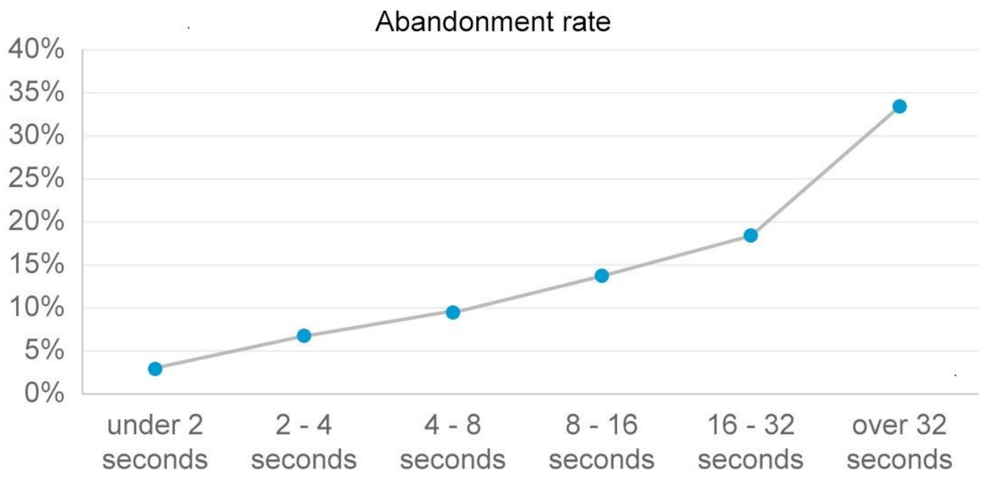

Figure 15 Graph of Website Abandonment v Time

Duration of a transaction	Likelihood to abandon
< 2 seconds	3.15%
2 – 4 seconds	6.78%
4 – 8 seconds	9.84%
8 – 16 seconds	14.07%
16 – 32 seconds	18.37%
> 32 seconds	33.91%

Table 9 Tabular Data for Figure 14

The two graphs are different as they represent behaviours in different contexts. The first one is viewing on-line videos and the second on is on-line transactions.

This can be factored in the analysis with a couple of important observations:

If our application process is on-line, then the information requested can be context sensitive and depend on the responses thus far. Only when a new piece of information is required, do we ask another question and bear the risk that a good customer abandons the transaction. If the rules require to reuse the same information, previously provided, that takes no appreciable time and we do not ask the customer for it again. For example, in the Lending Club decision tree we use FICO (a credit score invented by Fair Isaac and Company) several times, but only need ask once. This also impacts the decision to refer to the expert. If the potential abandonment costs of asking for more information, to feed another rule, is factored in, it may be better to refer prematurely if the net benefits of referring the decision are greater.

We have a similar situation on abandoned transactions as we do not know if they are good or bad. But we do know where they are in the rules and using the same process as above, we can estimate the probability that they are good and act accordingly.

The impact on the process is straightforward. We know the proportion of abandoned transactions as a function of the time (which we can relate to the number of questions we ask).

We can postulate that the abandonment process is the same as the survival process. It has infant mortality (where many people give up at the beginning), normal life (where the rate of abandonment is constant), and old age (where ultimately the process must end). This can be represented by the Weibull distribution (Cooray, 2006) with a probability density function (f) given by:

$$f(t) = \frac{\beta}{\lambda}\left(\frac{t}{\lambda}\right)^{\beta-1} exp-\left(\frac{t}{\lambda}\right)^{\beta} \qquad t \geq 0$$

$$f(t) = 0 \qquad t < 0$$

And the proportion that abandon in the interval of time [a, b] is given by:

$$prob(a < x < b) = \int_{a}^{b} f(u)du \;\; b \geq a \geq 0$$

And the cumulative density function (the proportion that have abandoned at time t) is given by:

$$F(t) = prob(0 < x < t) = \int_{o}^{t} f(u)du \;\; t \geq 0$$

$F(t) = 0 \qquad\qquad\qquad\qquad\qquad t < 0$

Integrating we get:

$$\boldsymbol{F(t) = 1 - \exp - \left(\frac{t}{\lambda}\right)^{\beta}} \qquad \textbf{Equation 1}$$

= Abandonment Rate (AR)

Where we have:

- Abandonment Rate is the proportion that have abandoned at time t
- β is the shape parameter the distribution. β<1 denotes that the occurrence of events is decreasing in frequency: β=1 denotes a steady state and β>1 indicates that the occurrence of events is increasing
- λ is the scale factor and is the mean value of t

If take rearrange and take the natural log of Equation 1 we get:

$$\ln(1 - F) = -\left(\frac{t}{\lambda}\right)^{\beta} \;\Rightarrow$$

$$\ln\left(\ln(1 - F)\right) = \beta(\ln t - \ln \lambda) = \beta \ln t - \beta \ln \lambda$$

If we plot $\ln\left(-\ln(1 - F)\right)$ (vertical access) against $\ln t$ we can see the fit. In Figures 16 and 17 the straight line indicates a good fit, the slope of the line is equal to β, and the intercept is $-\ln \beta\lambda$.

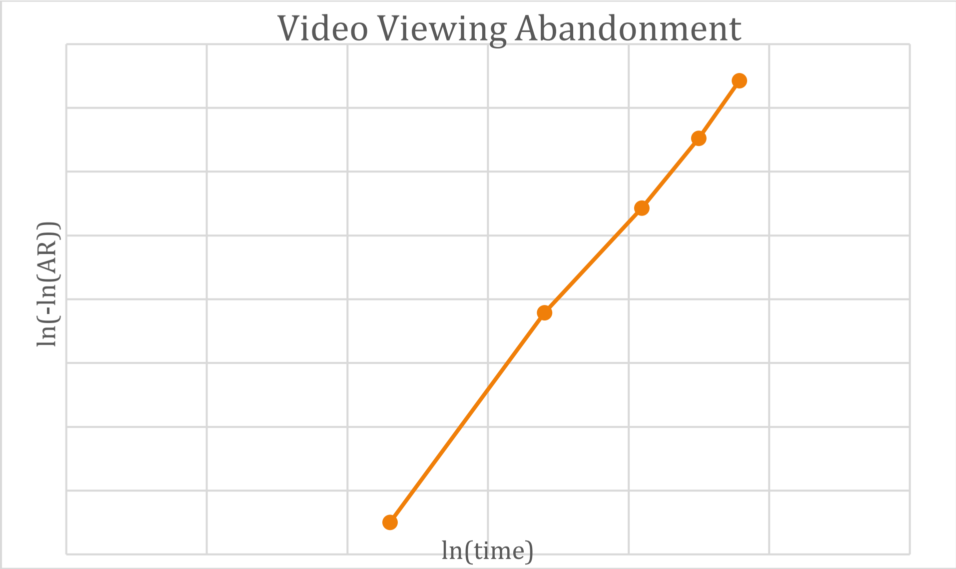

Figure 16 Weibull Plot of Video Abandonment

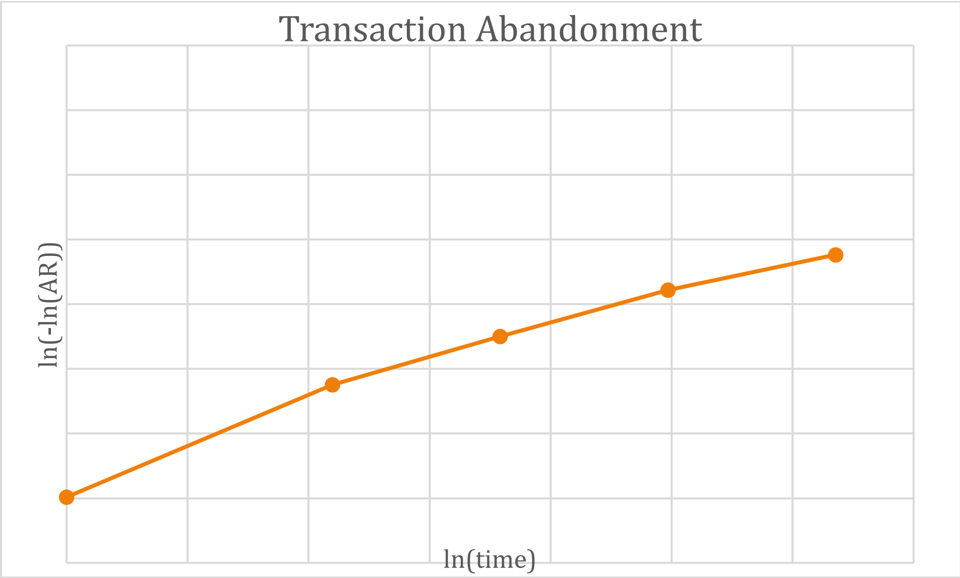

Figure 17 Weibull Plot of Website Abandonment

In both cases, there is a good fit with values of β and λ of 0.58 and 200, and 0.55 and 250. We would expect β to be less than unity which indicates that the rate of abandonment is decreasing at the beginning. Given that this is the only data we could find, it is surprisingly similar. That suggests that we could use the Weibull distribution to model transaction abandonment with parameters of 0.55 and 200 as a first estimate.

The final step is to substitute the number of attributes requested for time (t) in Equation 1. Assuming we have the data, we can calculate the time as a function of the number of attributes in the same way that we calculate the cost of referral.

$$t = t_I + kt_A$$

Where

t = total transaction time

t_I = initialisation of the process

t_A = time to respond to each request for information

6.4 Missing Data

We may have data on expert judgements and outcomes. This will allow us to fit logical regression models to predict $p(Eg)$ – the probability that the expert judges a case to be good - and $P(good|Eg)$ – the probability that a case that is judged to be good is, in fact, good - to the data and then calculate these quantities to each case. In the first case, we have the data on cases that are accepted and rejected. For the second case, we have data on the outcomes of cases, but only for those accepted. We do not have any data on the outcomes of the rejected cases. In the confusion matrix shown in Table 10 we have:

	Judged Good	**Judged Bad**
Good	a	b
Bad	c	d

Table 10 Confusion Matrix

We know a, c and b+d, but we do not know b or d because we don't have outcomes for cases that are rejected (judged bad). Setting b = 0 is somewhat optimistic as it assumes that every rejected case is bad. It is more reasonable to assume something like:

$$\frac{c}{a+c} = \frac{b}{b+d}$$

This equation accords with the idea that a judge will be equally wrong when judging a c. For example, if we have 60% of good cases and the judge gets 10% wrong, the results are shown in Table 11.

	Judged Good	**Judged Bad**
Good	54%	4%
Bad	6%	36%

Table 11 Error Model: a/(a+c) = b/(b+d)

If we take this approach, we have 4 equations for 4 unknowns, and we can solve them. We have:

$$d = c\frac{(1-a-c)}{(a+c)}$$

$$\Rightarrow d \ = (1-a-c)/(1+\frac{c}{a})$$

There is an issue when $p(good|Eg)$ is small and c/a becomes large, resulting in a small value for d a disproportionately large value of b. We are not interested in such cases, we just need to classify them as bad. Therefore, we modify the calculation to:

$$d = (1-a-c)/(1+\min(\frac{c}{a},1))$$

The results are quite reasonable and work well with the data set.

We could also assume:

$$\frac{b}{a+b} = \frac{c}{c+d}$$

The rationale her is that an equal proportion of good and bad cases will be judged wrongly. Using the equation above, we would have the situation in Table 12.

	Judged Good	**Judged Bad**
Good	54%	6%
Bad	4%	36%

Table 12 Error Model: b/(a+b) = c/(c+d)

In this case we have:

$$b = \frac{ac}{d}$$

$$\Rightarrow d + \frac{ac}{d} = (1-a-c)$$

This gives a quadratic equation for d:

$$d^2 - (1-a-c)d + ac = 0$$

Finally, we could set $b = c$ and have the results in Table 13.

	Judged Good	Judged Bad
Good	54%	5%
Bad	5%	36%

Table 13 Error Model: b=c

The first model is simple and is easy to solve, and reliable insofar as we know a and c and as long as $a + c < 1$ we can always find a solution. The second is more complicated and won't always solve. The solution relies on $(1 - a - c)^2$ being greater than or equal to $4ac$. This is not always true. If $a = 0.4\ and\ c = 0.4$ we have $(1 - a - c)^2 = 0.04\ and\ 4ac = \ 0.64$

The third is unreliable as there is no solution if $1 - a > 2c$.

The results of the first model are shown in Table 14:

P(Eg)	P(good|Eg)	a	b	c	d	P(good)
0.9	0.9	0.81	0.01	0.09	0.09	0.82
0.8	0.9	0.72	0.02	0.08	0.18	0.74
0.9	0.9	0.81	0.01	0.09	0.09	0.82
0.8	0.7	0.56	0.06	0.24	0.14	0.62
0.7	0.8	0.56	0.06	0.14	0.24	0.62
0.7	0.6	0.42	0.12	0.28	0.18	0.54
0.6	0.7	0.42	0.12	0.18	0.28	0.54
0.6	0.5	0.3	0.2	0.3	0.2	0.5
0.5	0.6	0.3	0.2	0.2	0.3	0.5
0.5	0.5	0.25	0.25	0.25	0.25	0.5
0.4	0.5	0.2	0.3	0.2	0.3	0.5
0.5	0.4	0.2	0.25	0.3	0.25	0.45
0.4	0.4	0.16	0.3	0.24	0.3	0.46

Table 14 Performance of Selected Error Model

6.5 Summary

There are several options for attribute selection but given the objective of a practical and reliable approach, the wrapper method should be preferable. Depending on the method we choose for rule building, dependent attributes may have been eliminated. If they remain, backward selection or feature correlated selection is an option.

With data on the outcomes of rejected cases missing, we must make assumptions on the ratio of good and bad outcomes for the rejected cases. Of the three intuitive methods, there is one method that works for all data sets, and we will use that later with the data in Chapter 8. But in the absence of data, there is no way to validate it.

In the two example data sets (or points) that we have found, transaction abandonment is a significant problem and should be factored into any attribute selection or rule building exercise.

7 RULE BUILDING

7.1 Introduction

In this chapter, we examine options for building rules that can be used for that set of problems that includes loan applications. The business problem has been described in Chapter 4, and the objectives of the rule system include:

- Getting the best financial outcome for the service business
- Identifying if it would be better to refer a decision to the expert

This requires a degree of self-awareness in the sense that the rules need to calculate the probability that the decision they have reached is correct. The rule system needs a model of the expert to assess the likelihood that he will make a better decision and one that justifies the additional cost of his involvement. The final choice of rule building must also consider performance (accuracy), degree of optimality achieved, and complexity.

We first revisit the objective function that we are attempting to maximise with the choices we make for the rules and make some observations around the benefits of reliability and simplicity that are useful in any practical method. Then we examine several potential ways to build the rules. We first propose a method to determine the optimal set of IF-THEN-ELSE rules from first principles. Then we examine the pros and cons of machine learning (decision trees and rule learning). Finally, we investigate the use of logistic regression and decision trees. We conclude that decision trees and logistic regression have some advantages in terms of simplicity, reliability, and practicality.

7.2 Objective Function

It is worth restating the objective function. We are aiming to build a set of rules that, when applied to the anticipate population, maximise the expected benefit (or profit) of good cases accepted, minus the expected cost (or loss) of any bad cases accepted in error. There are also costs incurred related to the amount of information requested and received, in terms of transaction abandonment and the human effort of processing information and making decisions. We can ignore the latter for the time being and concentrate on the potential methods to create rules.

As already stated, the rules that we are at liberty to vary are the CRA rules: computation, reasoning, and allocation. There are other rules, such as compliance, that do not have a bearing on our objective. For example, if it is against the law or a company rule to accept a class of customer, this customer will not be considered against any financial objective, they will be disqualified.

We have also observed that logical rules and regression have been identified as mechanisms used in human decision making, and they also have the advantage of simplicity and comprehensibility. As such, they represent a good model to work with. Other, more complex, techniques could be used, especially if they provide greater accuracy, but it may be impossible, or difficult, to understand why a decision was made.

7.3 Solution Architecture

The basic architecture is shown in Figure 18 and consists of a set of rules that decide whether a case should be accepted, rejected, or referred. We assume for now that we have data on expert judgement and the outcomes as in the LENS model. There are several ways we can create the rules that optimise the outcome for the organisation.

i. First principles; we use the training data and an optimisation algorithm to determine the tests within the rules and the outcomes.

ii. Identify those cases that would be better processed by the expert, identify them as such in the training set and then use machine learning to classify the cases as accept, reject, or refer

iii. Use logistical regression to model the expert and calculate the relative expected values of processing by the rules or the expert

We have also adopted the approach that, for a practical application framework, we restrict our attention to techniques that are available and as far as possible simple.

7.4 First Principles

Let us assume that we have a rule structure, and we need to determine:

- the parameters (such as the upper and lower limits in the logical tests and constants in any calculations) and
- choices (which rule do we use and potentially which logical or arithmetical operators do we use).

We can pose this as an optimisation problem where the objective function is the profit and the degrees of freedom are the parameters.

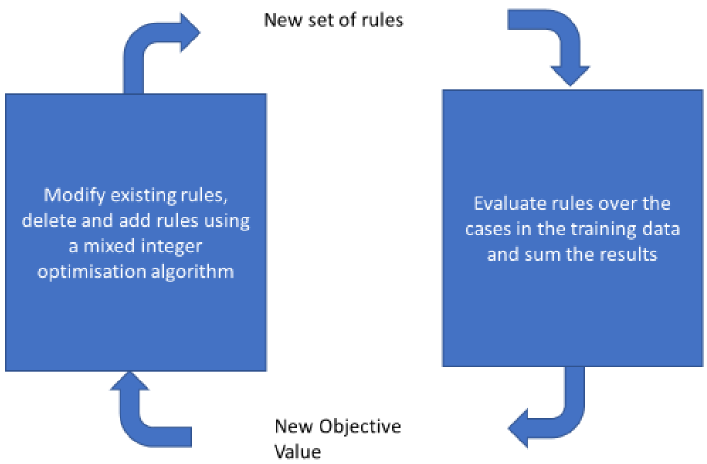

Figure 18 Rule Optimisation Algorithm

CRA rules include the value of parameters and logic within the rules. These pose an integer programming problem that is potentially non-linear as the objective function consists of calculations around profits and costs, and cumulative distribution functions (CDF) as identified in Chapter 5. For the type of problem we are interested in, CDFs are almost always non-linear.

Below we have a small example to illustrate the point which is a subset of a typical credit application problem where gender and income are two attributes of interest in the process:

$IF\ (\ x_1 = \ p_{13}\ )\ THEN$

$\quad IF\ (\ x_2 \geq \ l_{12}\ )\ THEN$

$\quad\quad IF\ (\ x_3 \geq \ l_{13}\ )\ THEN\ ACTION\ al_{123}$

$\quad\quad ELSE\ ACTION\ nla_{123}$

$IF\ (\ x_2 < \ u_{12}\ )\ THEN$

$IF\ (\ x_3 < \ u_{31}\ )\ THEN\ ACTION\ au_{123}$

$ELSE\ ACTION\ nau_{123}$

$ELSE\ IF\ (\ x_1 = \ p_{12}\ )\ THEN$

$IF\ (\ x_2 \geq \ l_{22}\ )\ THEN$

$IF\ (\ x_3 \geq \ l_{23}\ )\ THEN\ ACTION\ al_{223}$

$ELSE\ ACTION\ nal_{223}$

$IF\ (\ x_2 < \ u_{22}\ )\ THEN$

$IF\ (\ x_3 < \ u_{23}\ )\ THEN\ ACTION\ au_{223}$

$ELSE\ ACTION\ nau_{223}$

$ELSE\ IF\ (\ x_1 = \ p_{13}\ )\ THEN$

$IF\ (\ x_2 \geq \ l_{32}\ )\ THEN$

$IF\ (\ x_3 \geq \ l_{33}\ )\ THEN\ ACTION\ al_{323}$

$ELSE\ ACTION\ na_{323}$

$IF\ (\ x_2 < \ u_{32}\ )\ THEN$

$IF\ (\ x_3 < \ u_{33}\ )\ THEN\ ACTION\ au_{323}$

$ELSE\ ACTION\ na_{323}$

Where:

x_1 $is\ a\ choice\ variable\ that\ takes\ values$ $p_{11}, p_{13}\ and\ p_{13}$

x_2 $and\ x_3\ are\ real\ or\ integer\ variables$

l_{ij} and u_{ij} $are\ lower\ and\ upper\ limits$

al_{ijk} $, nal_{ijk}, au_{ijk}, nau_{ijk}$ $are\ equal\ to\ REJECT, REFER\ or\ ACCEPT$

The degrees of freedom for the optimisation problem are those listed above, and the objective function is the profit arising when the rules are applied to a data set containing past or expected cases. We have given some simple examples of this approach in Chapter 4, but to illustrate the point further assume that we have a customer i and the variables are:

x_{i1} $is\ gender\ and\ takes\ values\ male\ and\ female$

x_{i2} $is\ income\ and\ is\ a\ real\ number$

x_{i3} $is\ outcome\ and\ takes\ the\ value\ good\ and\ bad$

x_{i4} $is\ net\ profit$

$m\ and\ f\ are\ real\ variables$

$am, nam, af, naf\ \ are\ equal\ to\ REJECT\ or\ ACCEPT$

We build a model of the expected profit of the business as a function of the rule parameters and select values for the independent variables, populate the rules, take a set of cases and calculate the good and bad outcomes, and hence the profit, for the rules based on those variables.

Case No	**Gender**	**Income**	**Outcome**	**Profit/ Loss**
1	M	20,000	Good	40,000
2	M	25,000	Bad	-50,000
3	M	40,000	Good	70,000
4	M	45,000	Good	80,000
5	M	50,000	Good	90,000
6	F	20,000	Good	40,000
7	F	25,000	Good	50,000
8	F	40,000	Bad	-50,000
9	F	45,000	Good	80,000
10	F	50,000	Good	90,000

Table 15 Dataset of Cases

Table 15 is a simplified data set of historical cases and actual outcomes. We use this to evaluate the profit that would be generated had we used the proposed set of rules.

We then have a set of rules that look like:

$IF(x_{i1} = \ MALE)\ THEN$

$IF(x_{i2} > \ m)\ THEN\ ACTION\ am$

$ELSE\ ACTION\ nam$

$IF(x_{i1} = \ MALE)\ THEN$

$IF(x_{i2} \leq \mathrm{m})\ THEN\ ACTION\ alm$

$ELSE\ ACTION\ nalm$

$IF(x_{i1} = \ FEMALE)\ THEN$

$IF(x_{i2} > \ \mathrm{f})\ THEN\ ACTION\ af$

$ELSE\ ACTION\ naf$

$IF(x_{i1} = \ FEMALE)\ THEN$

$IF(x_{i2} \leq \ f)\ THEN\ ACTION\ alf$

$ELSE\ ACTION\ nalf$

So each time we set values for the independent variables: the real variables: $m\ and\ f$ and the binary variables: $am, nam, alm, nalm, af, naf, alf, nalf$ we have a different value of the objective function when these rules are applied to the data set. For example, the objective function (as a function of the variables above) looks remarkably like the rules:

$p = 0\ (set\ initial\ value\ of\ profit\ to\ zero)$

$FOR\ i = 1, 5$:

$$IF((x_{i1} = MALE)\ AND\ (x_{i2} > m)\ AND\ (am = ACCEPT))\ THEN\ p = p + \ x_{i4}$$

$$IF((x_{i1} = MALE)\ AND\ (x_{i2} \leq m)\ AND\ (nam = ACCEPT))\ THEN\ p = p + \ x_{i4}$$

$$IF((x_1 = MALE)\ AND\ (x_{i2} \leq m)\ AND\ (alm = ACCEPT))\ THEN\ p = p + \ x_{i4}$$

$$IF((x_1 = MALE)\ AND\ (x_{i2} > m)\ AND\ (nalm = ACCEPT))\ THEN\ p = p + x_{i4}$$

$$IF((x_{i1} = FEMALE)\ AND\ (x_{i2} > f)\ AND\ (af = ACCEPT))\ THEN\ p = p + x_{i4}$$

$$IF((x_{i1} = FEMALE)\ AND\ (x_{i2} \leq f)\ AND\ (na = ACCEPT))\ THEN\ p = p + x_{i4}$$

$$IF((x_1 = FEMALE)\ AND\ (x_{i2} \leq f)\ AND\ (alf = ACCEPT))\ THEN\ p = p + x_{i4}$$

$$IF((x_1 = FEMALE)\ AND\ (x_{i2} > f)\ AND\ (nal = ACCEPT))\ THEN\ p = p + x_{i4}$$

$REPEAT$

Note that we are only concerned with actions that result in acceptance. Erroneous acceptance (of a bad customer) and rejection (of a good customer) will both reduce the profit function by including an unnecessary loss or missing out on a potential profit.

We can then maximise this objective by manipulating the independent variables. The resulting ruleset is:

$IF(x_{i1} = MALE \quad (x_{i2} > 30{,}000)$

$IF(x_{i2} > 30{,}000)\ ACCEPT$

$ELSE\ IF(x_{i1} = FEMALE)$

$IF(x_{i2} < 30{,}000)\ ACCEPT$

$IF(x_{i2} > 40{,}000)\ ACCEPT$

$ELSE\ REJECT$

In this simple case, the objective function is not continuous; for example, if all other variables are equal and $m \rightarrow 25{,}000$ or $m \rightarrow 40{,}000$ we get a discontinuity. In a larger data set this would be moderated, but discontinuities would remain. The optimisation method to identify which rules to use and the limits on the attributes depends on the nature of the objective function and constraints. A linear problem may be solved by Linear Programming (Guéret et al, 2000). For non-linear problems there are a range of methods including direct search, gradient methods and methods that utilise information about second derivatives, such as Sequential Quadratic Programming (SQP) (Gill et al. 2005). There are also binary and integer variables (choices) to determine which require special consideration and these can be addressed with mixed-integer form of linear and non-linear programming, constraint programming (Apt, 2003) or evolutionary computing (Eiben et al., 2003). The choice between these methods depends very much on the nature of the objective function. Gradient methods are efficient, but they require the function to be continuously differentiable as a minimum; direct search is less efficient but imposes fewer conditions on the objective (Ali et al., 1997). Different methods may be appropriate depending on the rules and the underlying objective function relating to the business.

We can model any form of linear or non-linear functions and dependent variables, but with our problem, there are some serious difficulties:

i. The objective function will be differentiable as it is derived from cumulative distribution functions, provided that the cost functions are differentiable

ii. The theoretical objective function is differentiable, but the derivative is only right continuous (as identified in 5.4.3). That limits the application of methods that require gradient calculations.

iii. In practice, as identified above, the objective function would not be continuous

iv. Unless the objective is linear or convex, there is always a possibility of a local minimum (or minima).

v. There will be choice (integer) variables. There is no guarantee that non-linear integer programming problem will solve in a reasonable time, if ever (Schrijver, 1998).

Our objective is to create something useful and practical, and, for these reasons, we cannot rely on the first principles method. We can simplify the problem by linearisation, for example. But that step alone requires expertise that will limit its application.

7.5 Machine Learning

Business rules within the services sector are frequently concerned with making decisions that depend on the information provided (the characteristics or attributes) of an individual customer. In our case, we are making decisions about each customer on acceptance, rejection, or referral. As such, this is a classification problem.

There are many ways to solve a classification problem (decision trees, random forests, rule learning, clustering, and support vector machines, for example). Business rules are typically a series of logical tests on the available data – like a simple form of decision tree or the result of rule learning - or some form of a weighted scoring system. These have the distinct advantage of enabling the user to understand the process by which the rules decide. With more sophisticated methods (for example random forest or clustering) it is difficult – if not impossible – to understand why a decision was made. The other advantage with rules and decision trees is their ability to reach a decision without using all the data available in a context-sensitive manner. This is a significant advantage in our application over other types of machine learning that need all the data before a decision.

As such, we focus our attention – in this section – on decision trees and rule learning. In 7.7 we consider classification methods based on weighted scores.

According to the survey paper by Kotsiantis (2013), decision trees are sequential models, which logically combine a sequence of simple tests; each test compares a numeric attribute against a threshold value or a nominal attribute against a set of possible values. Given a population with various attributes that are known, decision trees can be used to classify members to determine attributes that are unknown. Each object belongs to a class and we can measure certain attributes, but we do not know the class, and when we know the class we can determine further attributes that we cannot measure. Rule learning is very similar.

For example, in medicine, there are things we can measure or know, such as age, sex, pulse rate and blood pressure but we want to determine the incidence of heart disease which cannot be readily determined. And, knowing that an individual is in a high-risk category for heart disease, we can determine other attributes, such as life expectancy. In a business situation, we do not know whether a customer will honour his obligations or how much he is prepared to pay; we must rely on information such as credit history, income, assets, etc.

Decision trees and rule learning are classification methods that are similar to business rules as they perform a sequence of logical tests on the attributes to arrive at a classification. They have some common features:

i. They both select the best attribute to branch on and then calculate a split based on a criterion such as maximising information gain or purity.
ii. They are robust and solve quickly, and there is publicly available software to train and analyse results
iii. Care must be taken with training to avoid overtraining and significant imbalance of the training set
iv. There are methods to cope with unequal misclassification costs which is relevant to the situation where the loss due to a false negative (rejecting a good customer) if not the same as a false positive (accepting a bad customer).
v. They cope well with non-linearity and can deal with situations where the value of an attribute can influence the impact of another attribute on the classification.

7.5.1 Decision Trees

Decision trees have typically used a set of members for which we know both class and attributes as a way of determining members for which we only know the attributes. We call this a training set, and the key assumption is that the relationships between class and attributes will not change. In essence, the past is a good guide to the future, or a sample is representative of the whole population.

Decision trees have some essential elements:

7.5.1.1 Optimising the Tests

Decision trees generally use an impurity criterion to choose the optimal test criteria, or in other words, which attribute to branch on next. (Timoveev, 2004).

One impurity measure of a set, t, is the Gini impurity, defined as:

$$I(t) \ = \ 1 \ - \sum_{i=1}^{n} p_i^2$$

Where p_i is the proportion of class k in each of the n sub-divisions. This is maximised when objects of the same class are evenly distributed ($p_k = \frac{1}{n}, 1 \leq i \leq n$) and minimised (i.e., zero) when all the objects of class k are in subdivision k (p_k =1 and $p_i = 0, i \neq k \ 1 \leq i \leq n$).

7.5.1.2 Pruning the Tree

Decisions trees should be as simple as possible. For example increased complexity can result in overtraining and poor prediction on new data sets. (Bramer, 2007). A pruning scheme is necessary which can be based on a minimum size for each node, either in absolute or relative (a percentage of the original set) terms.

We can also use cross-validation. This minimises a combination of the size of the tree and the misclassification error on randomly selected data sets. (Timofeev, 2004). In our case study we use this approach applied, for example to the J48 decision tree in Weka (Quinlan, 1993)

7.5.2 Rule Learning

This approach is similar in concept to decision trees (Fürnkranz, 2012). The idea is to classify elements based on a training set by constructing rules of the type:

$IF\ (Conditions) THEN\ c,$

where c is the class label

As in decision trees, variables can be discrete or real/continuous. This is like the categorical and ordered variables in decision trees, although integer variables are in essence both. The rule structure (Janssen, et al, 2010) created by rule learning is like that employed by business rules and is like this:

$IF f_1\ AND\ f_2\ AND\ \ldots.f_L\ THEN\ Class =\ c_i$

Where feature f_k, is a test that checks whether the element that we are classifying has the specified property or not. Examples of such features are:

$IF(attribute_k\ \in category_i)$ or

$IF(attribute_k\ \leq parameter_i)$

The distinguishing feature of rule learning is the rule structure, which is like decision trees and business rules that could be used for allocation and reasoning.

$IF\ (condition_i\ )\ AND\ IF\ \left(condition_j\right) THEN\ action$

$$IF\ (\ condition_i)\ OR\ IF\ \left(condition_j\right) THEN\ action$$

Rule learning is a vast field of research (Bundy, 1985), and we are more concerned with the form of the output (above) that resembles business rules. In the case study application, we apply propositional rule learning (Cohen, 1995).

7.5.3 Considerations with Machine Learning

7.5.3.1 Misclassification Costs

In many applications, there is a requirement to classify correctly. Still, the costs of misclassification may not be the same, and the cost of a false negative is different from that of a false positive. There are ways to compensate, for example, by weighting the training set in one direction or the other. This may appear to be crude, but with a two-class problem, the Gini impurity (Breiman et al., 1984) is insensitive to any modification. If we use the optimisation criteria instead of impurity, there is a potential for degeneration. It is also possible to modify the splitting criteria at the leaves.

7.5.3.2 Probabilities

Decision trees and rule learning algorithms do not directly calculate or estimate the probability of a case, or set of cases, being good (Kotsiantis, 2007), The only method is to inspect the leaf of the tree that contains the case. This can be prone to error, especially when there are few cases in a leaf. For example, if there are 5 cases in a leaf, a random change of +/- 1 case will give a 40% error in the estimate.

Recent research into rule learning has proposed methods to calculate probabilities (Kimmig et al., 2010). However, the authors concede probabilities are expensive to calculate and potentially infeasible.

7.5.3.3 Non-Linearity

Decision trees deal well with non-linear behaviour as they use a univariate search. Properly conducted, this will deliver the best split on a node by node basis. (Lin et al, 2014)

7.5.3.4 Dependency Between Attributes (Dependent Attributes)

There are situations where the category or value of an attribute influences the classification of another. A classic example is suicide in men that are classified as young and old, doctor and non-doctor, young men generally have higher rates of suicide, but if the man is a doctor, the situation is reversed. (Schernhammer, 2005) and (Gunnell, et al, 2003). With a decision tree, the branching on the subsequent variable takes this into account, and we can observe the reversal in the next layer. See Fig 19, below, for the behaviour in this case.

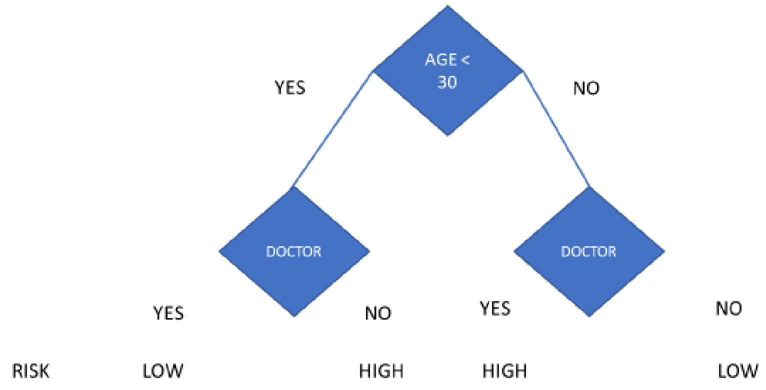

Figure 19 Behaviour of Decision Trees

7.5.3.5 Uncertain Classification

When the classification is unknown, we can eliminate those elements from calculations regarding impurity and then assign a classification based on the final node that contains them (Breiman et al., 1984). However, as we shall see in our case study data set, we need to be careful applying this approach to a whole distribution – for example, outcomes for accepted and rejected cases – when we only have a classification for the accepted cases.

7.5.4 Computation

Where classification is better on an axis other than the measured variables there is merit in using linear combinations of variables. This can be achieved during the test construction process and is commonly used in credit rating, through credit scoring. Computation rules can also be applied, such as ratios. For example, the ratios of loan to salary, or loan to valuation, are commonly used in mortgage applications.

7.5.5 Costs

Costs including execution costs, costs of tests and the potential (or expected) costs of misclassification can be included. There is also the notion of delayed costs (for example, tests that take time to carry out and where it would be impractical to wait for them). For example, in medical diagnosis, a doctor must be paid for a consultation, he may carry out additional tests (at additional costs) during the examination, and he may take blood samples for subsequent analysis. But when a blood sample is taken, it may make sense to gather further information, even though there may be information in the blood test result that may prompt further tests or action. This sort of problem involves balancing the cost of getting information and the accuracy of the classification. This has been considered by Turney (1995) and more recently, Freitas (2007). Chen (2009) takes this further and considers the problem of building a decision tree that deals with multiple attributes with an overall maximum cost, given by decision costs, test costs and the costs of misclassification.

Decision trees have also been applied to credit applications (Bensic, 2005) and (Galindo, 2000). In this research, the focus is the correctness of classification and how a decision tree can replace a human decision maker.

7.6 Application of Machine Learning

Decision trees and rule learning are classification methods that identify the next best attribute to branch on and use information gain (Gini impurity or entropy) to prevent degeneracy. They work well with non-linear functions and where there is a dependency between attributes. The main objective is to classify each case correctly. To apply these methods in our situation, we need to identify which cases should be referred. That requires the probabilities that either the rule or expert judgment is correct. With machine learning, we can calculate these probabilities using the numbers of good and bad cases in each node when we apply the decision tree or rules to a test set. We do not use any cases in the test set to create the tree or rules in the first place. The relative probabilities (will allow us to identify which cases should be referred. The remainder of the cases are left as good and bad, and we have three classes: accept, reject and refer.

For the cases that we accept and reject, there can be different misclassification costs. This is an example of cost-sensitive learning (Zadrozny et al., 2014) and in a two-class problem we would compensate by changing the proportion of good and bad outcomes in the training data. This change has the effect of moving the classification in favour of the class with a higher proportion. However, in our problem, we have three classes, and the misclassification is between reject and refer, or between refer and accept. We label a case in the training set for referral (as refer) based on the overall expected profits and costs (including misclassification costs). As such, for the final classification, we just require accuracy.

Finally, a major disadvantage of machine learning is that the identification of the case to refer (and hence the training data) will be dependent on the cost of the expert, and we will not be able to adjust this cost without repeating the learning process.

7.7 Logistic Regression

Logistic regression is a regression method that provides an estimate of the probability of an item belonging to one of the classes; as such, it can be used as a classifier. (Hosmer et al, 2013)). This is a major advantage over other forms of simple and multiple regression that simply produce a real number or integer without an estimate of probability. Such probabilities enable us to classify and calculate the probability that the classification is, indeed, correct.

Suppose we are classifying elements x_i $1 \leq i \leq n$ into two classes; good and bad, and each element has attributes a_{ij} $with$ $1 \leq j \leq m$.

We fit a linear regression model and for each element x_i we have an expression for the logarithm of the odds ratio as a function of its attributes:

$$l(x_i) = \ln p((x_i = good)/(1 - p(x_i = good))$$

$$l(x_i) = b_0 + \sum_{j=1}^{m} a_{ij} b_j$$

And

$$p(x_i = good) = 1/(1 + \exp(-l))$$

The main features of logistic regression is a direct calculation of probabilities (that an element is good or bad) on a case by case basis. This differs from using inference – from the results of application to the test set - with decision tree or rules. Direect calculation of probability is a significant advantage over decision trees and rule learning.

However, there are some other advantages and disadvantages:

7.7.1 Non-Linearity

Logistic regression is a linear method when we have real variables. Still, if we discretise the real variables, we can remove this limitation as each subset can have a different co-efficient. Methods such as Equal Width (of the bin) or Equal Frequency (of elements in the bin) are simple. Supervised methods are more complex, but Entropy Based Discretisation produces good results and is easy to implement (Mark et al, 2011). Essentially, we are looking subsets of the data (bins) with the lowest entropy subject to a stopping criterion based on Maximum Description Length.

For example, if we have two variables, x_i $0 \le i \le 2$ and we perform logistic regression we get an equation like:

$$\log \frac{p}{1-p} = b_2 x_2 + b_1 x_1 + b_0$$

Which is linear in x_1, x_2

But if we discretise and form $x_{11}, x_{12}, \ldots x_{1n}$ and $x_{21}, x_{22}, \ldots x_{2n}$ we get

$$\log \frac{p}{1-p} = \sum_{i=1}^{n} b_{2i} x_{2i} + \sum_{i=1}^{n} b_{1i} x_{1i} + b_0$$

Which is not linear x_1, x_2

The method estimates the probability that an element belongs to a class. As such, the results can be used for any objective function that depends on classification.

7.7.2 Dependent Variables

Logistic regression cannot represent the impact that one attribute has another in terms of the classification as each co-efficient is fixed and independent of the other attributes. Unlike decisions trees, it cannot split and then re-optimise the split again. This is an issue if we have such dependent attributes. It can be overcome by creating composite attributes, but this relies on them being identified, for example, by training a decision tree and looking for reversals.

By dependent variables, we mean that the value of one attribute influences the relationship between another attribute and the classification. For example, the risk of suicide in men is higher in young men, unless they are doctors in which case it is the reverse (see 7.5.3.4). If this is known, we can introduce a different discretisation scheme where we mix attributes. So instead of having:

$$A_k = \bigcup_{j=1}^{m} A_{kj} \ and \ A_l = \bigcup_{j=1}^{m} A_{lj}$$

As the discretisation of 2 dependent attributes we form a new discretisation such as:

$$A_k \cup A_l = \bigcup_{i,j=1}^{m} A_{ki} \cap \ A_{lj}$$

In the example above, we might have age in decades from 20 up until 60 (five categories) and doctor/non-doctor (two categories). Now we have ten categories:

20/D	**30/D**	**40/D**	**50/D**	**60/D**
20/ND	30/ND	40/ND	50/ND	60/ND

Table 16 Composite Attribute to Deal with Dependency

This enables logistical regression to apply the appropriate – higher – weighting to the shaded categories.

We group the cases within the union of one or more of the intersections defined by the discretisation process. To see this, let

- A_{ij} be the j^th discretisation of the attribute variable $x_i \ with \ 1 \leq i \leq n, 1 \leq j \leq m$
- Then the leaves or resultants of applying the rules mean that we create subsets:
- $A_{ij} \cap A_{ik} \ldots . \cap A_{il}$ for $1 \leq i \leq n, 1 \leq j, k, l \leq$ m

As previously identified, we compensate for unequal misclassification costs by optimising the cut off value for $p(good)$. Methods have been developed by (Bahnsen et al., 2014) for more complex credit scoring cost models, but this is not our focus for this research.

7.7.3 Efficiency

Logistic regression can support an efficient execution approach that consists of two elements:

i. Identify the optimal number of attributes required using feature selection (as discussed in 6.2)

ii. Identify the optimal order in which to process the information, potentially on a case-by-case basis

Note that, if we discretise the attributes, we have different coefficients in the expression for the log odds, and, as such, the order may differ.

7.7.4 Unequal Misclassification Costs and Sample Size

Strictly speaking, logistic regression is not a classifier. It produces (the logarithm of) the probability that a case is of a class. In our application, we only need two classes (good and bad, for example), but it can be applied to more than two. Logistic regression can made a classifier by applying a cut-off value which can be tuned to maximise the separation or reduce the misclassification rate. For these reasons, logistic regression is less sensitive (although not immune) to unequal sample size, and we can compensate for unequal misclassification costs by tuning the cut-off. For example, when the sample size and misclassification costs are equal, the cut-off is approximately 0.5. There is no formula to adjust this when we do not have equality, that is dependent on the nature of the data set.

7.7.5 Previous Applications

Logistic regression has been applied to credit scoring (Bahnsen et al., 2014), for example, which includes unequal classification costs. Like machine learning research, the focus is on replacing, rather than augmenting, the human.

His paper is somewhat unusual as it uses data on outcomes explicitly. However, it does not make the link with judgements, nor does it consider the outcomes for cases that were rejected.

7.8 Application of Logistic Regression

Applying logistic regression is like the application of machine learning. The regression produces an estimate of $p(good)$ and we can choose a cut-off that creates a classification function. Also, we can vary the cost of the expert as the calculation of the expected net benefit is embedded with the process of deciding which cases to refer, which cases are good, and which cases are bad. This is a significant advantage for the operations problem when the marginal cost of the experts is zero.

7.9 Conclusions

7.9.1 First Principles

The complexity and unreliability of non-linear optimisation is the main problem. The theoretical objective may be differentiable, but not continuously so, but is more likely to be discontinuous in practice. There is also no guarantee of a unique minimum, or absence of a local minimum. However, there are some advantages:

- The problem, if it solves, is the best possible solution. Decision trees and rule learning work attribute by attribute which may be sub-optimal
- In more complex cases, the participants and operators within the calculations may be included as optimisation variables. This negates the need for calculations to be pre-defined
- The existence – or otherwise – of rules can also be optimisation variables. This is another way to do feature selection; if the rule that uses an attribute is not required, neither is the attribute

Although the rules created may look like a decision tree there are differences:

- Decision tree building consists of a series of univariate optimisation problems, attribute by attribute, and uses an information gain or entropy objective function. The first principles approach uses the objective function that we are interested in, such as profit or revenue.
- Decision trees create different tests for each branch. First principals will only create values for tests that are in the proposed structure
- Constants and choice of logical or arithmetic operators are out of scope with decision trees which only choose the tests

7.9.2 Machine Learning

The main issues with decision trees and rule learning are the requirements to identify referral cases, a priori, and then build the tree or rules. If the costs change, the referrals change, and we must rebuild. In addition, determination of the purity (or quality) in each leaf by inspection is subject to error. However, the structure is easy to understand; the process naturally elicits the most important information first and can reach a decision without using all the available data, and can deal with dependencies.

7.9.3 Logistic Regression

There are some disadvantages compared to machine learning; we must discretise to deal with non-linearities and be careful with dependent variables. However, the process is simpler as we only have one step, we can vary the costs without having to repeat the regression, and we can calculate probabilities directly on a case by case basis.

7.10 Summary

We can rule out first principles on the grounds of complexity and unreliability. Machine learning deals naturally with non-linearity and dependent variables. It also processes the most important data first and can reach a decision without all the data. It is, therefore, a good choice for the business rules.

Logistic regression requires some care and thought if we have non-linearity and dependent variables, but the process of application is simpler, and we can adjust the cost of the expert 'on the fly'. It can, therefore, be used to build the rules and has the advantage of calculating probabilities directly. As such, it also a good choice for modelling the performance of the human expert.

In practice, we also consider the accuracy of classification, which is problem dependent with each technology. Subject to this, we conclude that machine learning and logistic regression can be used for the rules, and that the human expert is best modelled using logistic regression.

Table 17 summarises the key points of this argument.

Method	**First Principles**	**Decision Trees**	**Rule Learning**	**Logistic Regression**
Simplicity	No	Yes	Yes	Yes
Speed	Slow	Fast	Fast	Fast
Non-linear	Yes	Yes	Yes	With discretisation
Affected by imbalanced data sets	No	Yes	Yes	Partly
Objective	Economic	Purity	Entropy	Error
Output	Rules	Tree	Rules	Probability
Deals with differing costs	Yes	Not well	Not well	Yes
Dependent attributes	Yes	Yes	Yes	No
Case by case costs	No	No	No	Yes
Calculates probabilities	Indirectly	Indirectly	Indirectly	Directly
Context sensitive	Yes	Yes	Yes	Possible

Table 17 Comparison of Rule Building Techniques

8 LENDING CLUB

8.1 Introduction

To test the theory and methods that have been developed requires real data. We need datasets with attributes, judgements and outcomes of a size that enables optimisation or machine learning and independent validation. There is little publicly available data of this nature; Lending Club is an exception, and it has published large data sets containing this information.

In this section, we describe the current process used by Lending Club and the nature of the data used for determining whether a loan should be accepted or rejected, and the subsequent results data. We also identify where there is scope for introducing business rules and automation to augment the current process. We assume but do not know that the current process will have some rules, and we can deduce some of them. But in any event, having data on judgements and outcomes, we can test out hypotheses on real data.

Lending Club no longer makes the data publicly available on the site, but the data can be found on Kaggle (2020).

8.2 Process

The Lending Club data consists of two types of file:

rejected applications with data including:

- State
- First three numbers of the Zip Code
- Amount of loan ($)
- Credit Score (originally FICO and later on Vantage)
- Debt to income ratio (for existing loans excluding mortgage) DTI
- Employment length in years

Loan statistics with additional information,

- Loan status information
    - Application approved or not
    - Fully paid, current or charged off (charged off means delinquent and unlikely to be repaid)
- Annual income
- Interest rate
- Purpose

Unfortunately, the data sets do not match the application process. The initial application process requires the following information.

- Name
- Address
- Income
- Amount
- Purpose

Applicants are then screened and a large number of rejected. Those that make it through are then asked for more information, and some are subsequently rejected, and the rest are set up with an interest rate that is directly related to their credit score.

Note that FICO and DTI are significant in so far as they both contain information about the potential borrower’s credit experience. FICO derives the score by considering factors such as credit history. This includes how much was borrowed and whether all repayments were made, in full, on time. It is proprietary, but the general idea is disclosed. DTI is defined as current monthly debt repayments divided by gross monthly income. This explains the upper limit of 30% as typically lenders do not approve applications when debt levels are already high.

There are effectively three parts to this process that could be modelled.

- The Initial Screening
- The more detailed screening (document and credit checks)
- The loans that subsequently that are either completed or charged off

Some rules are employed in the initial screening. For example, some potential customers are automatically rejected:

- Applicants for loans in excess of $40,000
- Applicants with a DTI greater than 30
- Applicants with a work status of n/a (assumed not working)

8.3 Data Preparation

In both data files, there is some data that is obviously wrong or not helpful.

- Work status n/a appears in the application data. If this is assumed to be not working and the applicant would be rejected, so these were removed.
- Residence reported as NONE. There is currently OWN, MORTGAGE, RENT and OTHER (assumed to be living with parents, relatives). There are a few NONE and these have been removed as during the analysis they gave rise to a very high co-efficient, reflecting the characteristics of the few loans with that label.
- High values of DTI have been combined into one set as they are clearly erroneous (the loan would not have been accepted in the first place).
- Information on working status in the loan data is assumed to be the status at the time of application. For example, if working status was 1 year at the time of application it would be 3 years after 2 years of the loan (assuming no change of job).
- There is information in finance blogs (Lend Academy, 2011) about how Lending Club reduce borrowers' credit score for larger loans. This is contradicted in the loan status data (where there are large loans with credit score A). As this cannot be substantiated, it has been ignored.

We decided to focus on the loan status data, which refers to the final approval stage because we have data on outcomes rather than judgements. There are approximately 45,000 loans in the data set, and we have data on accepted and rejected, and outcome data for the accepted loans.

Analysing this should give some insights into the ability to decide on the application automatically and/or to refer the decision to the expert

8.3.1 Data Analysis

The data set of accepted and rejected applicants (in total 45,000) that we used is analysed in Table 18:

Item	Loan Value	Loan Term	FICO Score	Employment Length	Income	DTI
Min	4800	36	633	0	1440000	5.34
Max	35000	60	990	15	2000	29.9
Range	30200	24	357	15	1438000	24.56
Average	11000	41	804	5.9	76400	13

Table 18 Data Analysis

Also, we had:

- 48% of applicants had a mortgage
- 45% were renting
- 7% owned their home outright
- 29% of applicants were rejected
- 15% of accepted applicants eventually defaulted

8.4 Initial Screening Application

The first part of the Lending Club process is an initial screen of applicants using a limited number of attributes.

We examined the CRA problem, which is how many cases should be given to the human expert. We assume, initially, that the expert is 100% accurate as we have no data on outcomes, but in the next section – where we do have data on outcomes – we will model the competence of the expert. We also assume that the potential profit (of taking on a good customer) and loss (from taking on a bad customer) are equal to 50% of the average loan value. We do not have the exact numbers, and these are reasonable considering the level of interest rates (around 10%), the term (36 or 60 months) and that defaults can occur at any time. These numbers are simple inputs and can easily be varied.

Initial feature selection showed that the address attributes (State and Postcode) and Loan Amount were of no value to the classification. We use the wrapper method and greedy stepwise algorithms in WEKA (Eibe et al., 2016).

Weka provides access to visualisation tools and algorithms for data analysis and predictive modelling, together with graphical user interfaces for easy access to these functions. Weka has typical data mining tasks, such as extraction, data pre-processing, clustering, classification, regression, visualization, and feature selection.

The specifications for feature selection (taken from Weka) are:

WrapperSubsetEval:

Evaluates attribute sets by using a learning scheme. Cross-validation is used to estimate the accuracy of the learning scheme for a set of attributes.

For more information see Kohavi et al., (1997).

GreedyStepwise:

Performs a greedy forward or backward search through the space of attribute subsets. May start with no/all attributes or from an arbitrary point in the space. Stops when the addition/deletion of any remaining attributes results in a decrease in evaluation. Can also produce a ranked list of attributes by traversing the space from one side to the other and recording the order that attributes are selected.

A simple 5 level decision tree was built using the WEKA Workbench with the remainder of the attributes.

The specification for the decision tree is:

weka.classifiers.trees.J48

Class for generating a pruned or unpruned C4.5 decision tree. For more information, see Quinlan (1993).

The attributes are shown in Table 19 (below):

Attribute	**Description**
FICO	Credit score from Fair Isaac and Corporation
DTI	Debt to income ratio (%) for existing loans (excluding mortgage)
EMP	Number of complete years with current employer

Table 19 Attributes for Initial Selection

This decision tree is shown in Figure 20.

FICO < 802.5

| EMP < 0.75 : 0 (236800/649) [118014/306]

| EMP >= 0.75

| | DTI < 24.99

| | | DTI < 4.05 : 0 (25206/344) [12687/190]

| | | DTI >= 4.05

| | | | FICO < 751.5 : 0 (44825/4009) [22450/1906]

| | | | FICO >= 751.5

| | | | | FICO < 752.5 : 1 (520/126) [256/49]

| | | | | FICO >= 752.5 : 0 (3716/911) [1760/445]

| | DTI >= 24.99 : 0 (30260/28) [15403/26]

FICO >= 802.5

| FICO < 819.5

| | FICO < 803.5

| | | DTI < 2.54

| | | | DTI < 0.39 : 0 (17/2) [9/4]

| | | | DTI >= 0.39 : 1 (60/27) [30/11]

| | | DTI >= 2.54

| | | | DTI < 30.02 : 1 (905/27) [486/27]

| | | | DTI >= 30.02 : 0 (14/0) [8/0]

| | FICO >= 803.5 : 0 (864/0) [429/0]

| FICO >= 819.5

| | FICO < 835

| | | FICO < 820.5 : 1 (952/23) [508/6]

| | | FICO >= 820.5 : 0 (40/0) [25/0]

| | FICO >= 835 : 1 (10491/2) [5271/2]

Figure 20 Decision Tree for Initial Screening

At each (of the fourteen) leaves of the decision tree we have information on the type of node (0= bad, 1 = good) and the composition (#correct, #incorrect). The numbers in square brackets are the cross-validation set implemented automatically by Weka. For example, on the first leaf we test for FICO < 802.5 and then EMP < 0.75. We determine that the node is bad with 236800 correctly classified (as bad) and 649 incorrectly classified. The problem now is to determine whether it is worth giving these cases to a human judge, and incurring a cost, or simply rejecting all of them automatically and accepting the cost of rejecting a relatively small number of good cases.

If we set the cost of an initial determination at 0.5% of the average loan, we find that it is optimal to assign 77% of the cases to the rules engine, and the gross profit increases by 1.8% over that obtained by using the human judge on all cases (shown in Table 20 below). With a determination cost of 0.25% of the average loan value, the allocation remains the same, with an increase in gross profit of 0.6%. The break-even point is reached when the cost of a determination is 0.134% of the average loan value. The profit figures below are $1,000's.

	OPTIMAL CASE LOAD		PROFIT
Leaf	Machine	Human	Increase
1	355769	0	15616
2	0	38427	0
3	0	73190	0
4	0	951	0
5	0	6832	0
6	45717	0	2419
7	0	32	0
8	0	128	0
9	0	1445	0
10	22	0	1
11	1293	0	78
12	0	1489	0
13	65	0	4
14	15768	0	910
Total	418634	122494	19028
	77.36%	22.64%	1.81%

Table 20 Results for Initial Screening

The diagram shows that if we take the machine determination for leaf one, the net saving over giving the cases to the human judge is $15,616,000. The same applies to leaves six, ten, eleven, thirteen and fourteen. For the remainder, it is better to give the case to the human, with no net saving.

8.5 Final Selection Process

Credit application checking has been extensively researched in many areas, particularly artificial intelligence (AI) with decision trees and rule learning (Lahsasna, 2010). By and large, the emphasis is to be as good as (or better) than the human decision maker. We have chosen this example purely because we can establish some realistic data on judgements AND outcomes.

This is an example of the LENS model (Brunswik, 1985) where models are built of judgements and actual outcomes. These models use multiple linear regression. The estimates of the outcome and judgment are linear combinations of the attributes With some mild assumptions around independence, it is possible to relate the various correlation coefficients into an elegant equation so that the performance of the human judge can be estimated across a collection of cases. Unfortunately for us, this is not sufficient as we require accuracy on a case by case basis. We need to estimate the probability that the human judge will determine that a case is good (or bad) and the probability that this is, indeed, correct. Logistic regression was chosen in this case as it produces an estimate of these probabilities. We have used the Weka package (Eibe, 2016) for analysis.

We use the same technique to estimate the probability that a case is good or bad. We need this to calculate the probability that the expert is correct; the probability that a case is good or bad given that the expert judges it to be so. Therefore, the rules are based on probabilities and expected values derived from logistic regression. But provided we can estimate the accuracy of the expert judgement on any case as a function of what we know about that case (that is, the attributes) there is no limitation to the type of rules that are employed. We make the conditional decision that a case is good or bad and then decide whether that decision should be implemented directly or that there is greater expected value in referring that to an expert.

To address the problem, we need to work out the difference between accepting the decision of any automatic decision mechanism, for example, a decision tree, and that of giving the decision to a human. For example, we may anticipate a better (more accurate) decision, if we give this to the expert, but we must balance the cost. We recognise that there are many potential models of decision support, and this could include help and advice, provided interactively. Such systems have a place, but the focus (and novelty) in our approach is that we are calculating the probability that the rules and the expert will be correct (or wrong) in their assessment. This is then used to calculate the expected benefit of implementing either decision, if the expert decision is better, the difference in expected benefit should exceed the cost of the expert.

8.5.1 The Selection Process

The initial screening process has already been analysed using a decision tree to determine how much could be automated. The results indicate that about 80% of assessments could be automated with a net benefit; the overall expected profit would go up slightly. We used a decision tree as did not need to estimate the accuracy of the expert; the assumption was that the expert had perfect judgement. This was not an unreasonable assumption for the purposes of that exercise, but in this section we use data on judgements and outcomes We could have used any method to create the rules, but we chose logistic regression because of simplicity, and their unique ability to calculate the probability that a case is good or bad, on a case by case basis and the probability that a rule outcome or judgement is correct or incorrect, again on a case by case basis. To use to use other types of rule, for example decision trees, we can use the contents of the leaves of the tree to estimate probabilities. This is more complicated and potentially prone to error when there are few cases in a leaf.

Analysis of the detailed, loan status, data shows that the expert is not entirely accurate and that our previous assumption of complete accuracy would not reflect reality. Some 15% of loans become delinquent and were either late or charged-off (assumed never to be repaid). This is a considerable cost, and we cannot, therefore, assume that passing the decision to the expert will result in an accurate assessment.

The continuous data were discretised as follows:

- Loan amount in $2,500 steps up to $40,000 and above
- Income in $25,000 steps up to $300,000 and above
- DTI in increments of 2.5 up to 30 and above

These increments were arrived at by gradually reducing the increments until there was no improvement in classification accuracy.

Using the utility in Weka, we randomly chose 2/3 of the data to fit regression models of the expert and outcomes, leaving 1/3 for a test set. We do not need a validation set as Weka supports cross-validation automatically by partitioning the training set into 5 or 10 subsets and using all but one to train and one to validate. The process is then repeated by removing another subset and training and validating again.

We can model the human or expert using the confusion matrix shown in Table 21.

	Judged Good	**Judged Bad**
Actual Good	a	b
Actual Bad	b	d

Table 21 Confusion Matrox

And given any sample size, S, we have:

$a + b = Sp(good)$

$c + d = Sp(bad)$

$a + c = Sp(Eg)$

$a + b = Sp(Eb)$

Where

Eg and Eb denotes when expert classifies the case as good or bad

Firstly, we carry out a logistic regression on the decision of the expert. This will give us an estimate for p(Eg), the probability that the expert decides an applicant is good.

Logistic regression works quite well for the accepted cases but works less well for the rejected cases. This is not surprising as the rejected cases in this set are somewhat borderline, given that there has already been a first-round rejection. This also suggests that there may be other factors influencing the final acceptance decision; the expert manages to distinguish between cases that present as similar based on the data available.

We then carry out a logistic regression on the good and bad outcomes for those loans in the data set (the accepted loans). This should give us an estimate of the probability of default given that the loan was accepted p(good|Eg). Note that this is a conditional probability and is NOT the same as p(good).

Here, unlike most previous research, classification accuracy is not the objective. We are looking for a way to determine which cases can be automatically classified and which ones could be given to the expert.

To test the efficacy of this approach we have applied the following process to the test set to determine what the decision would be, and then calculate the costs and benefits. This first analysis was to determine if there was any merit in this approach, not necessarily to fine-tune the process. Nor are we checking the accuracy of the regression; this has already been determined by cross-validation. What we are doing is running through a process on past data to indicate how well it could work in practice.

There is further potential to fine-tune and validate with other data. However, at this stage, we are more interested in whether it works at all.

The procedure is outlined below. Essentially, the machine branches on whether it calculates the case to be good or bad, based on the estimate of p(good). It then calculates the expected value (outcome and processing costs) of passing the case onto the expert.

For each case:

Calculate $p(Eg\ and\ good)$ and $p(Rg\ and\ good)$ using the logistic regression parameters for each case

Calculate $p(Eg\ and\ bad)$ and $p(Rg\ and\ bad)$ in the same way

Calculate the net gain for the rules and expert

Expected gain from using the rules decision:

$Rgain = p(good\ and\ \ Rg).profit - p(bad\ and\ Rg).loss$

Expected gain from referral:

$Egain = p(good\ and\ Eg).profit - p(bad\ and\ Eg).loss - cost$

For this analysis we assumed that potential profit and potential loss are equal to the size of the loan, which results in the cut off parameter for the machine being set to 0. If the profit and loss were not equal, this imbalance would be reflected in a different cut off value. The processing costs are a function of the amount of information, for example, a familiarisation time and a time proportional to the amount of information. We estimated this cost to be 5% of the loan value.

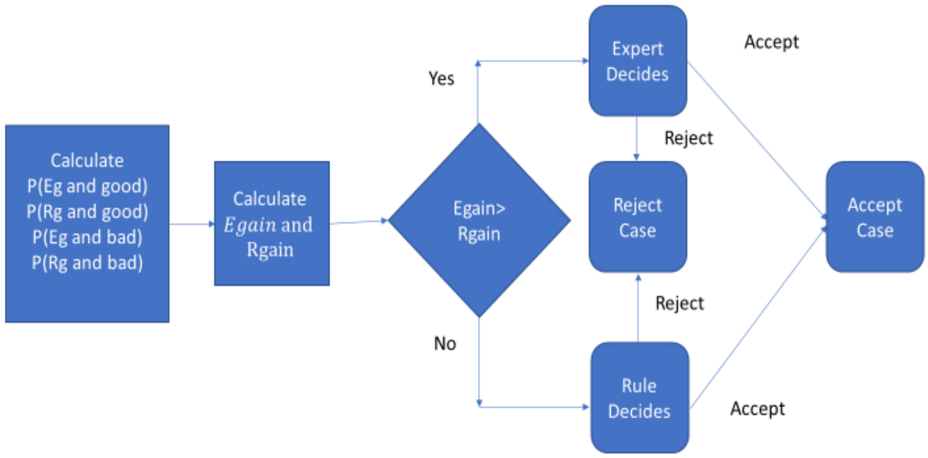

Figure 21 Proposed Rule Execution Flowchart

There are several potential sophistications possible:

- Potential profit and loss are different:

  The potential profit is a function of the size of loan, terms and interest rate

  The potential loss is a function of size, term, interest rate and estimated (weighted average) time of default

- We allow for customers that drop out based on the quantity of information initially or subsequently requested
- We allow for improved processing and judgement accuracy based on additional information
- Experts have different costs and judgement accuracy

We decided that these were not possible with data available to us, but that this sort of data would be available to an organisation in the business of credit approval.

The other problem that can be addressed with this approach is the operational problem. This is distinct from the design problem where we decide on the overall proportion of cases that need expert assessment and, hence, the number of experts. Once we have employed a specific number of experts, we can modify the rules to make the best use of the experts that are now available.

Applying the rules that were developed to a subset of the data that was not used to create them yielded some interesting results About 30% of the cases were deemed better given to the expert, the remaining 70% left to the machine. The overall benefit, as a percentage of the loan book, was between 4%. This is like the previous result that assumed that the expert was perfect. It sent 80% of the cases to the machine and made a 2% improvement. The cases with unknown outcomes impacted the benefit of this approach. We had to assume that, if the machine accepted a case that the expert had rejected, then the outcome would have been bad. That’s not necessarily the case, and it does rather reduce the benefit. Even so, this approach (under the assumptions made) has some benefit. Overall, there were potential benefits equivalent to about 4% of the loan book, which given that Lending Club is now at about $24Bn, is worth at least further analysis.

Another point to note is the significance of labour cost saving. Relying on expert labour is a risk for any business, there is always variability that needs to be managed, and the business is not scalable. If we can automate a large proportion of transactions and understand the trade-off between labour costs and net loan value, variations in applications (up and down) can be better accommodated.

Note that the assumption that the case with an unknown outcome is pessimistic, particularly that this data set does not include applications that are rejected without further analysis. If we assumed that only half of these cases are bad, with the other half being good, the benefit increases to 7%. We could legitimately say that the potential benefit is between 4% (no unknown cases are good) and 10% (all unknown cases are good). Note that there are no unknown cases when they are passed to the expert; we know already that his or her judgement is to reject.

Table 22 below shows the first results of applying this approach to the data set, with all amounts in 1,000’s.

	EXPERT	MACHINE	EXPERT
	ONLY		
CASES	100%	68%	32%
GOOD	$123,849	$95,291	$27,872
BAD	$22,207	$15,829	$5,997
UNKNOWN		$1,480	
EXPENSES	$7,302		$1,958
NET	$94,340	$77,982	$19,918
TOTAL	$94,340	$97,900	
SAVING		$3,561	3.78%

Table 22 Results of Optimisation (Dormer, 2018)

Other researchers have used Lending Club data. For example, (Emekter et al, 2014) analyses the outcomes and concludes that loans with lower credit grade and longer duration are associated with high mortality rate. Serrano-Cinca develops a logistic regression model to predict defaults. The grade assigned by the lending site is the most predictive factor of default. Still, the accuracy of the model is improved by adding other information, especially the borrower's debt level. Chang et al. (2015) conclude that results on data from Lending Club (LC) indicate the random forest-based method outperforms the FICO credit scores as well as LC grades in the identification of good borrowers. Increases in investment returns from better classification are estimated at 50%. This is comparable with our figure of 3.78% when investment returns are around 7.5%.

These research results are broadly in line with our calculations and conclusions that decision trees and logistic regression using the available data are better than FICO for predicting default. This research does not explore the combination of rules with the current decision making process.

8.6 Summary

The results indicate that there is potential benefit in combining business rules with human decision makers/experts. The results are better than either rules or experts alone. For the initial screening we chose to use a decision tree. For the final selection, we have used logistic regression to categories cases and to estimate the accuracy of the expert. We could have used a decision tree for the initial screening, but getting a similar result using a different approach is encouraging. This is even more significant when we consider the limitations of the data, and that only accepted cases have a known outcome.

This business process has some extraordinary features. Firstly, it processes people, and the different characteristics of those people are fundamental to the process. The differences in the applicants cannot be abstracted away to allow us to deal with “average” people. Indeed, on any given day, the set of customers who request to be served may be quite different in their characteristics. Secondly, the benefit from those customers chosen to be served is not a function of the number of chosen customers but is intimately linked to the process by which they are chosen. Thirdly increasing the resources allocated to the process will increase cost but may not yield an increase in the number of customers chosen to be served.

The method we have created works best when we have data on outcomes, as the rules can be built using this data. In this case, we should expect benefits from efficiency (reducing the cost of human effort) and accuracy (reducing the impact of human errors and bias). If we do not have data on outcomes, there is still scope for improved efficiency, noting that the rules can emulate the human expert and therefore reduce cost. If we have information on the average accuracy of the expert, this can also be included easily.

We have demonstrated and quantified a practical method for optimising the rules and determining the number of cases that are processed automatically and by the human expert, based on a historical data set with some simplifications and a fixed set of attributes. The method essentially looks at the value that can be obtained by stopping at any stage and accepting the decision that would have been made or progressing to the next stage and accepting an additional cost in return for a higher expected value.

9 VARIATIONS AND EXTENSIONS

9.1 Introduction

In this chapter, we extend some of the initial ideas to cover the situation where the data requested, or the order that the data is requested, can vary based on the utility of that data and the responses obtained to previous requests. In the first instance, we examine the two-stage approach where a limited amount of data is requested and then decisions are made on what to do next. Lending Club use this approach and there is good reason for this:

- The data suggests that a high percentage of transactions are abandoned, even after a short time
- More data may make the evaluation more accurate, but with high levels of applications (that do not qualify) the costs are higher

In this chapter, we investigate the situation where a limited amount of information is requested initially – like the Lending Club process. The rules use this to decide what to do next.

The options are:

- Decide without further information
- Request further information
- Refer to the expert

The working assumption is that once referred, the expert decides whether to decide or request further information.

If the rules decide to request further information, there is then the option for the rule to decide or to refer. This has been explored previously in Chapter 8. We now focus on the performance of the rules and the expert on a reduced set of information.

We show that there are benefits to this approach, due mainly to the impact of transaction abandonment. This impact can be a reduction in potential customers of 25% or more, and the losses can outweigh the benefits of additional information in a large proportion of cases. There are different approaches to this issue; we can request a block of information and then decide whether we ask for another block, or we can identify the best option and end point on a case by case basis. Also, we can order the collection of attributes for all cases, or on a case by case basis.

In this chapter, we build up the problem, adding complexity step by step. First, we limit the problem to two sets of attributes, an initial set and then an optional additional set. The rules are configured to decide, on a case-by-case basis:

- whether one or both sets of attributes are required, and,
- whether the rules or the expert makes the final decision.

We then allow the rules to request attributes, one-by-one, again on a case-by-case basis, with the option for rule-based or expert decision making. Finally, we allow the rules to determine the next best attribute on a case-by-case basis, with decision making as above. This is the most complex situation with the maximum freedom and potential benefit.

9.2 Data Selection

The selection of the initial variables is more complex than a typical feature selection problem. We can use Weka feature selection (using the filter, wrapper, or correlation-based filter methods) to determine the optimal set of attributes. We can also use Weka to determine the ranking, and, alternatively, find the largest weights in the logistic regression expression. Unfortunately, none of these uses the objective function that we are interested in, which is the net profit after allowance for transaction abandonment. The correct method is to use the wrapper with the objective function specific to the problem.

For any set of rules, we are required to model additional paths and costs, including:

- The loss from applicants that abandon (which is a function of the number of data items requested)

- The profit from applicants that are correctly approved – the loss from applicants that are incorrectly approved
- The cost of processing (assumed to be a function of the number of data items requested)

Using this method, we find that the best subset is the first 6 attributes:

FICO (credit score)

Verification (whether documents have been provided and checked)

Home Ownership

DTI (debt to income ratio)

Purpose (of the loan)

Income

The benefit of further attributes is effectively cancelled out by transaction abandonment so the most logical first set of attributes for the rules to work on are these. We can then look at the potential for referral after that.

9.3 Variations of the Problem

There are several variations, with increasing levels of computational complexity.

At the simplest level we have:

- A predetermined subset of attributes is requested/collected. These are the same for all cases and would generally be determined by feature selection or some form of ranking exercise
- The system decides to adopt the decision of the rules, continue with the rules or refer to the expert
- If the decision is to continue, the remaining attributes are collected in one block and processed by the rules
- If the decision is to refer, the expert decides whether to decide or request the remaining attributes

The process can be made more sophisticated by removing the limitations and assumptions above.

- On a case by case basis and as each attribute is collected, we can choose to accept (the rule), continue or refer.
- Obtaining additional information on an attribute by attribute basis, rather than in a fixed block
- We can vary the order that attributes are collected on a case by case basis

9.4 Re-estimating Expert Performance

When we reduce (or consider reducing) the number of attributes we assume that we have data on the attributes and associated judgements and outcomes for the complete data set. We then work out a new set of rules on the reduced set. That is not a serious problem as we just refit and validate. The expert judgement is a bit more difficult as we need to simulate what the expert would have decided on the reduced attribute set. This illustrates another benefit of using logistic regression to model the expert. If we reduce the number of attributes, we can effectively model the impact on the expert judgement by eliminating that attribute from the expression and modifying the intercept to make up for the removal of the attribute from the overall expression.

When we restrict the number of attributes we collect, the outcomes are the same, we just need to refit the rules using the reduced set of attributes. Previously we had the logistic regression function

$$l_{OUT}(x_i) = \sum_{j=1}^{m} a_{ij}b_j + b_0$$

Where $l_{OUT}(x_i) = \ln p/(1-p)$

and $p = p(outcome\ x_i = good)$

We just refit outcomes with m-k attributes and get

$$l(x_i) = \sum_{j=1}^{k} a_{ij}b_j + b_0$$

However, we cannot use this process for the expert. His judgements were formed using the complete set and, unlike the outcomes, we anticipate that his judgement would differ with fewer attributes. Effectively we need to model the impact of using only the information contained in the k attributes that remain. In this case, we fit the original judgements against the original data (where l is the log odds function):

$$l_{ACC}(x_i) = \sum_{j=1}^{m} a_{ij}b_j + b_0$$

Where $l_{ACC}(x_i) = \ln p/(1-p)$

and $p = p(judgement\ x_i = good)$

We separate the expression:

$$l_{ACC}(x_i) = \sum_{j=1}^{k} a_{ij}b_j + \sum_{j=k+1}^{m} a_{ij}b_j + b_0$$

We then reset the constant term to reflect removal of the attributes and their overall contribution across the population, summing across all the elements of $x_i, 1 \leq i \leq n$. In this way, we are effectively removing the individual contribution of each element and then adding back the average contribution. Just removing the relevant terms in the equation above would bias the judgement one way or the other, unless their sum across the whole population was zero.

We have:

$$\sum_{i=1}^{n} l_{ACC}(x_i) = \sum_{i=1}^{n}\sum_{j=1}^{k} a_{ij}b_j + \sum_{i=1}^{n}\sum_{j=k+1}^{m} a_{ij}b_j + nb_0$$

To keep the LHS the same we set:

$$\acute{b}_0 = b_0 + \sum_{i=1}^{n}\sum_{j=k+1}^{m} a_{ij}b_j / n$$

then we have

$$\acute{l}_{ACC}(x_i) = \sum_{j=1}^{k} a_{ij}b_j + \acute{b}_0$$

To do this, we need to create several models:

- A model of the outcomes, using a reduced set of attributes
- A model of the correct acceptances – p (judged good AND good) - using all the attributes and then modify it only to include the reduced set

- A model of the incorrect acceptances – p (bad AND judged good) using all the attributes and modified as above
- Fit the Weibull distribution for abandoned transactions using data on the current data set (n data items) and calculate the value for the reduced set (k data items). Note that the Weibull distribution requires two parameters, and we may only have one data point (based on the current operation). If this is the case, we can use a default value for β of 0.5 and calculate λ. If this is somewhat arbitrary, we could simply use linear interpolation.

The method can be described as follows:

For each case we calculate:

$p(good\ AND\ Eg|k)$

$p(bad\ AND\ Eg|k)$

$p(good\ AND\ Rg|k)$

$p(bad\ AND\ Rg|k)$

We also calculate abandonment rate, R, and the cost C, as a function of the number of attributes

$R = W(k)$

$Cost(k) = ck + f$

Where f is a fixed cost for the expert to open a case and c is an additional cost for each attribute. The form of this function does not need to be simple if further data on costs were available. We could also allow for differences between experts in costs (speed) as we could for other areas of performance, such as accuracy if we had sufficient data.

We then calculate expected net benefits (gain) given the first k attributes:

Expected gain from using the rules decision:

$$Rgain = W(k)[p(good\ AND\ Rg|k).profit - p(bad\ AND\ Rg|k).loss]$$

Expected gain from referral:

$$Egain = W(k)[p(good\ AND\ Eg|k).profit - p(bad\ AND\ Eg|k).loss - cost(k)]$$

We then need to calculate the benefit of additional information (either by the rules or the expert). This can take 3 forms:

i. We request a predetermined set of attributes and have the option to request the remainder
ii. Fixed order, where we ask for additional attributes in a predetermined order, but we have the option to request additional attributes, one by one
iii. Variable order, where we calculate the next best attribute based on the increase in the net benefit, starting from one attribute that has been determined by feature selection and ranking

The last method is the most appealing (on the basis that an increase in the degrees of freedom can only equal or improve the result of an optimisation problem). Still, there is the complication that the gain from the rules and expert will not be the same for one or more of the attributes. To deal with this issue, we need to add each additional attribute to the equations above and identify the best attribute to ask for next, for both rules and expert. We then repeat the process until the gains of having all the attributes, in the best order (for rules and experts). We also observe that the first referral incurs the fixed cost element. This means that the case will stay with the expert unless the marginal benefit of using the rules for subsequent attributes is greater than that of the expert.

9.4.1 Attributes Processed in Two Blocks

In this situation, we have two sets of information. We decide on a pre-determined set that is initially requested based on, for example, a feature selection exercise or experience. Each case comes with that information and the rules are required to decide whether to accept or reject (using the initial information), to request further information or to refer. If a case is referred, the expert has the option to decide or to request further information. We use the model of the expert to predict his behaviour. We are not concerned at present with the order of the attributes as they grouped, and the initial and additional attributes are assumed to be and processed as a package.

Using the notation above, we have a set of k attributes that we request initially and the option to request another m-k.

We can calculate:

$Rgain(k)$: the expected net gain of using the rules with k attributes

$Rgain(n)$: the expected net gain of using the rules with n attributes

$Egain(k)$: the expected net gain of referral with k attributes

$Egain(n)$: the expected net gain of referral with n attributes

And then, for each case we have the following logic:

$IF\ Rgain(k) \geq Rgain(m)\ AND$

$Rgain(k) \geq Egain(k)\ AND$

$Rgain(k) \geq Egain(m)$

Denoted in shorthand by:

$IF\ Rgain(k) \geq Rgain(m), Egain(k), Egain(m)$

Then the expected net gain from using the rules with k attributes is greater than the other options and use the decision of the rules and stop. However (using the same shorthand notation):

$IF\ Rgain(m) \geq Rgain(k), Egain(k), Egain(m)$

It is better to continue with the rules and request a further block of attributes to a total of m. Then:

$IF\ Egain(k) \geq Rgain(m), Rgain(k), Egain(m)$

We refer to the expert with k attributes in the expectation that he decides without further information. We could also, of course, advise him that he ought to do this. Finally:

$IF\ Egain(m) \geq Rgain(m), Rgain(k), Egain(k)$

We refer to the expert in the expectation that he decides with further information. We could also, of course, advise him that he ought to do this.

9.4.2 One by One Attributes/Fixed Order

In this scenario, we have the option to request further information on an attribute by attribute basis. Using the notation that we have defined, we need to consider:

$$GR = \max_i Rgain(i)\ for\ 1 \leq i \leq k$$

$$GE = \max_i Egain(i)\ for\ 1 \leq i \leq k$$

Then we can deduce:

$IF(GR \geq GE) and\ GR = Rgain(k)$

we stop and use the rule decision

$IF(GR \geq GE) and\ GR = Rgain(m)\ with\ m > k$

We continue with the rules

$IF(GE > GR)$

We refer to the expert

It is important that we calculate the gains all the way to the end of the process as the optimal order of asking for information for the rules will be different to that for the expert.

9.4.3 One by One Attributes/Optimal Order

As previously noted, decision trees and rule learning naturally organise the tests in an optimal order and can reach a decision without using all the available attributes. Then, at each node, we can determine whether the rules or the expert is the best choice for the decision. With logistic regression, this is more complex, and we need to calculate the gains and compare.

To address this problem, we form an ordered set for the gains for the rules and the expert. For each remaining attribute, we calculate the gain and select the attribute with the highest gain as the next one. This process is repeated until all the attributes are used. The optimal (attribute) order may be different for the rules and the expert and will differ between cases. Given the computer power available now it is quite realistic to calculate this as the cases arrive. Still, if we categorise the real attributes, there will be a finite number of combinations and we could pre-compute and simply look up the appropriate values. For example, if we have n attributes and a maximum of m categories for each, we have a maximum of $2mn$! calculations (since we could have different values for the rules and expert).

$Rgain(i)\ for\ k \leq i \leq n$ to be the gain from the rules for the (optimal) rule using the attribute in position i

$Egain(i)\ for\ k \leq i \leq n$ to be the gain from the rules for the (optimal) rule using the attribute in position i

We then set:

$GR = \max_i rgain(i)\ for\ k \leq i \leq n$ and

$GE = \max_i egain(i)\ for\ k \leq i \leq n$

9.5 Results

9.5.1 Test Set

We select a set of cases at random from the validation set to evaluate the methods that we have developed. The data set was heavily skewed towards acceptance, so we removed some accepted cases using bagging to even up the data set to have an equal number of good and bad outcomes. This is consistent with a symmetrical cost of misclassification and creates a better model. The test set, therefore, was consequently smaller. There are approximately 4,500 cases in the evaluation and validation set from an original data set of 45,000. For the accepted cases we have data on outcomes, for all other cases we estimate the probability using the method in 6.3 and assign an outcome as good when the probability is greater than or equal to 0.5. Otherwise we assign a bad outcome.

9.5.2 Lending Club Base Case

For the base case we use all the attributes and categorise the outcomes of the rejected cases. We use the model to decide whether to accept, reject or refer. For those that the rules accept, we use the actual outcome. For those that are referred, we use the actual expert decision. In this way, we calculate the anticipated net benefit of the system. Below we show the original situation, with the expert alone, and the new situation, with rules and expert, combined. We see that about 80% of the cases can be decided by the rules alone, with an increase in net benefit using the cost ratio

VALUE	GOOD	BAD	COST	NET
ORIG	1722	293	224	1205
NEW	1831	383	50	1398
CASES	**EXPERT**	**RULES**		
ORIG	4498	0		
NEW	1000	3498		

Table 23 Benefits of Referral

Table 23 gives the results. The ORIG line is with the expert processing all the cases, and the NEW line is with the expert and the rules. GOOD refers to the value of good cases accepted, and BAD refers to the cost of bad cases accepted. COST is the referral cost (which for the rules is zero). NET is GOOD-BAD-COST, the overall benefit to the organisation.

This shows that there is a better net result of 1,398 if we combine the rules with an expert. Previously the expert achieved a net profit of 1,205 by accepting 1,722 good cases and 293 bad cases, and with a processing cost of 224. The combined result reduces processing cost to 50, accepting 1,831 good cases and 383 bad cases.

9.5.3 Blocked Attributes

In this case, we select a subset of attributes that we use first and then another subset next.

We have applied the process identified above where we request 6 attributes, and then decide whether to:

i. Ask for more data and continue with the rules
ii. Refer to the expert (with the model suggesting that he will ask for more data)
iii. Use the current determination of the rules
iv. Refer to the expert (with the model suggesting that he will use the current data)

To solve this, we set up an initial problem, using nine attributes (based on the earlier feature selection exercise using Weka in 9.2), estimating the outcome of the unknown cases (those that were rejected) and allowing for abandoned transactions. We then remove the contributions of the three removed variables (to get back to the minimum number of six attributes that were calculated using the wrapper method and the specific objective function, again in 9.2) and add back the average value to the constant term in the expression for the log odds. We then compare the maximum values of $rgain$ and $egain$ for the two cases (six attributes and nine attributes), and we also allow for the abandonment factor and cost factor (both of which increases with the number of attributes requested). We identify the maximum value and its location to determine whether the case should be decided:

- After six attributes by the rules
- After six attributes by the expert (referral now)
- After nine attributes by the rules (continue with rules)
- After nine attributes by the expert (refer and recommend to seek more data)

This process showed a significant improvement in the objective. The improvement came from cost reduction and a more accurate estimation of the expected outcome for the unknown cases. The data above values each case at unity (for potential profit and loss) with an expense ratio of 5%. If we then reduce the number of attributes to 6, taking away the lowest ranked attributes (employment, income, and existing debt to income) we get this result.

	ATTRIBUTES	GOOD	BAD	COST	NET
Rules	6	0	0	0	
Expert	6	0	0	0	
Rules	9	1380	104	0	1276
Expert	9	451	279	50	122
Total		1831	383	50	1398

Table 24 Nine Attributes Requested

Table 24 shows the results of asking for nine attributes with either an accept, reject or referral at the end of the process. The final row is the same as in table 23.

	ATTRIBUTES	GOOD	BAD	COST	NET
Rules	6	1105	194	0	911
Expert	6	913	200	63	650
Rules	9	0	0	0	0
Expert	9	0	0	0	0
Total		2018	394	63	1561

Table 25 Six Attributes Requested

Table 25 shows that the results of asking for only six attributes is much better due to the reduction in transaction abandonment. Note that we had already determined this using the wrapper method of feature selection in 9.2 using the full objective function and allowance for transaction abandonment.

	ATTRIBUTES	GOOD	BAD	COST	NET
Rules	6	699	162	0	537
Expert	6	588	135	40	413
Rules	9	390	35	0	355
Expert	9	319	136	31	152
		1996	468	71	1557

Table 26 Option of Six or Nine Attributes

Table 26 shows the impact of asking for either 6 or 9 attributes on a case by case basis. The three additional attributes do not add any value overall to the profit function.

This is not an interesting result and, as we will show later, the reasons are:

- Assuming equal misclassification costs is probably not what happens in practice

- The impact of transaction abandonment effectively blunts the utility of further data
- The three least significant attributes add very little value

To illustrate the point, we can remove transaction abandonment and repeat the process. Table 27 shows the results of asking for six attributes, and Table 28 shows the results of asking for nine.

	ATTRIBUTES	GOOD	BAD	COST	NET
Rules	6	1105	97	0	973
Expert	6	913	100	63	750
Rules	9	0	0	0	0
Expert	9	0	0	0	0
Total		2018	197	63	1723

Table 27 Six Attributes Without Abandonment

	ATTRIBUTES	GOOD	BAD	COST	NET
Rules	6	0	0	0	
Expert	6	0	0	0	
Rules	9	1725	130	0	1595
Expert	9	564	349	62	153
Total		2289	479	62	1748

Table 28 Nine Attributes Without Abandonment

If we now choose, for each attribute, the best option of current, refer or continue, we get the results in Table 29:

	ATTRIBUTES	GOOD	BAD	COST	NET
Rules	6	699	81	0	618
Expert	6	588	68	40	480
Rules	9	490	17	0	473
Expert	9	319	68	31	220
					1791

Table 29 Option for Six or Nine Without Abandonment

Most of the incremental benefits (over the previous results adjusted for cart abandonment) can be explained when we look at the abandonment rate. We have used a mean of 200 seconds and beta value of 0.55. This distribution is rather pessimistic, but it is consistent with the public data. There is other evidence that transaction abandonment levels can be around 27% (Tarasofsky, 2008). In our data set, using the current determination, rather than accept the additional cost of cart abandonment, was a definite advantage. Table 30 show transaction abandonment as predicted by the Weibull distribution.

Data Item	1	2	3	4	5	6	7	8	9
Time	3	6	9	12	15	18	21	24	27
Remain	91%	86%	83%	81%	79%	77%	75%	73%	N/A

Table 30 Transaction Abandonment following Weibull

9.5.4 Attribute by Attribute

In this case, we remove all but one of the attributes and then add back in the order of significance. The most significant attribute is the FICO score. This is not unreasonable as this is a composite attribute based on factors including how well previous loans have been serviced. That is of some considerable interest here. For each case we calculate the maximum value of $Egain$ and $Rgain$ and determine the optimal position and decision-making process, considering transaction abandonment.

For this problem, we start with the most significant attribute and make the calculations of $Egain$ and $Rgain$ for each, for all the attributes in the set order of calculation. We identify the maximum value, and that determines the decision maker (rules or expert) and the number of attributes that we collect for that case. We factor in the abandonment rate and information processing costs, as above.

	FICO	**VER**	**HOME**	**DTI**	**PURP**	**INCOME**	**ALL**
Good	2043	2068	2058	2006	2034	2018	1831
Bad	517	527	520	473	442	394	383
Costs	14	25	31	33	47	63	50
Objective	1512	1516	1507	1500	1545	1561	1398

Table 31 Feature Selection

Table 31 show the progress of the feature selection as attributes are added. The objective stays around 1500 and the increases to a maximum of 1561. The decrease at the point where we use all the attributes is due to the impact of additional costs of processing and transaction abandonment.

9.5.5 Optimal Order, Position and Decision Maker

This is the final refinement of the method that considers order as well as position and decision maker. We start with the most significant attribute – in this case, FICO – and then determine the next best attribute based on the increase of the maximum value of $Egain$ and $Rgain$. This is done for each case until we have used all the attributes. The benefit of more attributes should be a better decision; the disbenefit is the reduction in the probability that a case will complete the transaction.

We use a forward selection approach to the problem for two reasons. Firstly, we include all the attributes; there are no stopping criteria that would be affected by dependent attributes (which is the reason that forward selection fails). Secondly, the working assumption is that dependent attributes have been identified and steps taken to reformulate the problem to make it more amenable to logistic regression.

Note that, if this has been done the attributes in question will be considered together, as pairs, with associated abandonment and information processing costs. Starting with the most significant attribute we add each remaining one, in turn, using the criteria of the best overall organisational outcome (generally profit, but it could be revenue) generated by using that pair of attributes.

The results are shown below. Transaction abandonment has been set to zero to illustrate the process better. Otherwise, as shown above, most of the decisions will be made using only the first attribute.

	FICO	VER	HOME	PURP	ALL
Rules	387	450	383	243	1051
Expert	924	112	3	141	655

Table 32 Attribute by Attribute Feature Selection

Table 32 shows the optimal distribution of the decisions between rules and experts based on the maximum value of expected net gain. For example, the first column shows that 387 cases are best sent to the rules and 924 cases are best referred, both with just FICO as a data point.

9.6 Summary

There are 3 variations on the interactive optimisation problem:

- Consider attributes in 2 (or more) blocks
- Consider attributes one-by-one, in a fixed order
- Consider attributes, one-by-one, in an optimised order

With the Lending Club Data, each refinement creates a better solution, and each refinement is better than the previous method that used every attribute.

These results are essentially due to the high impact of transaction abandonment. The model suggest a peak at around 25% with nine attributes requested. This accords with measured data. It is worth remarking that transaction abandonment is not a loss, per se, more a lost opportunity. As such it is easier to ignore than a loss that is crystallised, such as an unpaid loan.

The data, and relationships, point to a much more considered approach where opportunity cost is compared to loss when we decide how much information is required. The model suggests that, in many cases, a decision can be made on much less data and that any subsequent loss due to a loan going bad cane be compensated for by additional good customers who would otherwise abandon.

10 Operational Optimisation

10.1 Introduction

Up to now, we have been solving the business rule optimisation problem, and the result is a set of rules that determines how each case should be processed. A by-product is the optimal number of experts that are required, and the proportion of cases should be allocated to the rules, and the proportion allocated to experts given the caseload and case-mix that we anticipate. In an ideal world, caseload would be constant, and case-mix would reflect the training data. In practice, this would not happen, and experts would be, at different times, idle or overcommitted. If we go to the trouble of business rule optimisation, we should have a way to make maximum use of available resources which the organisation pays for them whether they are used or not.

There is also the possibility that the relationships between the attributes and the outcomes change. For example, if interest rates rise, different people may ask for loans and/or more loans will default.

This chapter develops a method to adjust the rules so that we can give the experts the cases that will be most advantageous to the organisation, and avoid giving them so much work that the system slows down and queues build up. We also consider the medium to long term problem of retraining or re-optimising the rules as the case-mix changes.

10.2 Expert Resource Requirements

There are two reasons why we may have an excess or shortfall of experts for any situation:

- We may have a different caseload
- We may have a different case-mix

In any event, we need a strategy to adjust the rules to make the best use of the experts that we have allocated to the task. During the initial optimisation phase, we did not limit the number of experts, we just solved the optimisation problem, and the optimal cost (that included the cost of the expert intervention) was one of the results. That cost translates to a resource. Now we have that resource we have an issue of allocation; what cases do we allocate within the limits of the available resources.

10.2.1 Solving the Problem

We have a queue of cases that are processed through the rules and, at each point we have (or calculate) whether the rule decision should be accepted, we should refer or ask for further information.

Previously we have assumed no limit on the number or input of the experts. Indeed, the objective was to use as many as we needed, provided their input was cost-effective. Now – by implication – we have decided that, on average, a certain percentage of cases will be referred.

The nature of the problem is such that we have incurred the cost for the experts employed and marginal cost of their input is now effectively zero. So, where we would have deducted a cost for expert judgement, it is now free but now, we have a fixed number of experts that can process, at most, a fixed number of cases.

In the previous chapter, we had this solution for the nine-attribute problem. Table 33 shows the results.

VALUE	**GOOD**	**BAD**	**COST**	**NET**
ORIG	1722	293	224	1205
NEW	1831	383	50	1398
CASES	EXPERT	RULES		
ORIG	4498	0		
NEW	1000	3498		

Table 33 Nine Attribute Problem

Let assume that an expert can process twenty cases per day and the figures above represent a week's worth of cases. As such, we require ten experts to process a thousand cases, and the weekly cost of an expert is five.

We recalculate $Egain$ and $Rgain$ without the costs and denote these as $Emgain$ and $Rmgain$ to signify we are using marginal costing. By marginal costing we mean the cost varies (typically as a function of work to be done) rather than the cost that is incurred regardless: the so-called fixed cost. For example, in a transport problem a salaried driver and owned vehicle are fixed costs; we incur them regardless of how much work we need to do. Conversely, the cost of a rented vehicle with a casually employed driver is marginal; we can rent more or fewer vehicles and employ more or fewer drivers depending on how much work we have to do. In our case, if we assume that the experts are salaried employees, their cost is fixed and not a function of the caseload or the case-mix. So, we can ignore their cost in the operational sense and just focus on the expected benefit of allocating cases to either the rules or the experts.

Expected gain from using the rules decision using k attributes:

$$Rmgain = W(k)[p(good\ AND\ Rg|k).profit - p(bad\ AND\ Rg|k).loss]$$

And expected gain from referral:

$$Emgain = W(k)[p(good\ AND\ Eg|k).profit - p(bad\ AND\ Eg|k).loss]$$

For each case we rework the calculations in Chapter 10 using these formulae. For cases where it would have been best to use the expert, this will still apply as we just have a higher value for $Egain$ (it increases by $W(k).cost(k)$). But for the other cases, some of them may result in a referral. And for all cases, we calculate the highest possible value of

$$Net\ gain = \max_k(Emgain - Rmgain)$$

For each case

Once we have identified all these cases, we order $Net\ gain$, with the highest first. Then we take as many cases that the experts can process.

For example:

- If the caseload reduces, the expenses stay the same and the efficiency of the operation reduces. The optimal response is to allocate more cases to experts where $Emgain > Rmgain$ but previously, we had $Rgain \geq Egain$.
- If the caseload increases, the available experts stays the same, and we risk losing the capacity to process the cases at the correct rate. More cases will be allocated to the experts than we anticipate, and a queue will build up that reduces the number of cases that will be approved. We have already identified time as a cause of transaction abandonment in the application process; it is logically a factor in the approval process where, if we take too long, the customer loses interest or goes elsewhere. The optimal response is to take some cases away from the experts that have the least impact on profitability. These are the cases where $Emgain - Rmgain$ is the smallest.
- If the case-mix changes, we follow a more general procedure, which applies equally well to the situations above. We take a batch of cases and determine those that, ordinarily, would require expert judgement. We compare that number to the expected number (the expected proportion times the number of cases in the batch), and we either have a de facto increase or decrease in workload. We then apply one of the two methods above.

10.2.2 Lot-Sizing & Queuing

The lot-sizing problem (Sox et al, 1999) is common in manufacturing where larger lot sizes have lower manufacturing cost but higher inventory requirements. The same thinking can be applied to the service sector, and Deb et al. (1973) investigate the batch servicing problem where costs are weighed against the size of the queue time or waiting time. In our case study, we have the same considerations around the size of the queue. However, if the queue is small, the proportion of cases requiring an expert will be more variable and adjusting the rules with a smaller number of cases will be more difficult. Also, the queue will build up if the caseload, or case mix, requires more experts and reduce, and disappear when there is too little work for experts. In this case, we can change the rules to reduce the expert workload, as above. If the opposite occurs, and we have experts with nothing to do and no queue, we categorise the cases as:

- $Emgain > Rgain$ – allocate the case to the next available expert

- $Rmgain \geq Emgain$ - no benefits in allocating to the expert

  We do not have any data on the impact of waiting time for a decision from the expert or request for further information on customer behaviour. A reasonable assumption would be that there is an impact.

  The lot-sizing problem here is more complicated than service optimisation, where the aim is to serve the customers within a given time when the arrival times are not constant but represented by a Poisson process. This is the classical queueing problem. Here we have that problem, but we also have another factor which is the ability to allocate the most appropriate cases (i.e., the ones with the most marginal benefit from expert assessment) to the experts.

  For the queuing problem, Deb et al. (1973) propose that when the queue length, x, exceeds a certain maximum, M, we serve min (x, S) customers where S is the service capacity. M and S depend on a range of cost and service time factors. Due to lack of data, there is no point computing M and S; suffice to say it cis possible. Also, we are only interested in the number of cases in the queue where Emgain > Rmgain, and their arrival rate. So, to provide a given service level and waiting time, the number of experts must be the maximum of the optimal number E (from the rule optimisation) and S (from service optimisation).

  For the lot-sizing problem, we need to know the distribution of Emgain. We have (since Emgain > 0 for the cases in question):

  $$prob(Emgain < x) = \int_{o}^{x} em(u)du = eml$$

  $$prob(Emgain > x) = \int_{x}^{\infty} em(u)du = emh$$

  $$expected\ profit(Emgain > x) = \int_{x}^{\infty} em(u)udu = p(x)$$

  Where em is the probability density function of Emgain.

  So, to give the cases that earn us at least x to the experts, we need a queue of size

  $\boldsymbol{Q} \geq \boldsymbol{E/emh}$ to create a profit $\boldsymbol{p(x)}$ **Equation 2**

  To see this, suppose 10% of cases have a profit of x or more. To be able to allocate these to 5 experts, we would require at least 50 cases in the queue on average.

Equation 2 enables us to relate queue length (and waiting time) to overall profit, allowing the optimal queue length to be calculated.

So, to calculate the optimal number of experts and queue length.

- We choose the number of experts as the maximum of S and E and thus calculate the number of experts required, $E_{ACT} = \max(S, E)$ and the maximum queue length M
- Using E_{ACT} and Equation 2 optimise queue length (Q) W.R.T profit and waiting time with the (service) constraint that $Q<M$

The analysis below is necessarily simplified due to lack of data. But an organisation could quite easily collect the data required and carry out the optimisation based on these principles.

10.2.3 Results

We tested these procedures on a batch of 200 cases and calculated the number that required expert judgement. We used the simple process where we requested all 9 attributes, rather than adding further complexity, but the same approach will work with any level of complexity.

We selected 900 cases at random (a day's worth of cases) from the test set and calculated how the expert and rules would best decide the cases. This caseload, optimally, would require 200 expert decisions at a daily cost of 10. We assume that the budget is 10, for ten experts.

We then assume that the budget is reduced by 50% redirect those cases where the impact of the expert is the least useful (in terms of the expected gain) as shown in Table 33. We apply the process identified above and order the cases by the first column, below, shows the cases and the optimal expert allocation. The second column shows the impact of not addressing 100 cases and the resulting reduction in the net figure. The final column shows the impact of allocating only 100 cases to the expert and the rest to the rules. The rules themselves do not make as good a decision, but overall, the net benefit is better – as we would expect – but the capacity is maintained. The results are given in Table 34.

	Expected	**Actual**	**Optimal**
Cases processed	900	800	900
Good	366	315	365
Bad	77	50	74
Cases referred to expert	200	100	100
Cost	10	5	5
Objective	279	260	286

Table 34 Impact of Reducing Experts Available

Table 34 shows the impact of reducing the number of experts. There are 100 cases left over for referral, reducing the number of cases processed from 900 to 800. If we now modify the rules, we process 800 cases using the rules and the other 100 by the experts. Unlike the previous situation where there were an arbitrary 100 cases not processed, we now have the 100 cases that contribute the greatest value when processed by an expert.

If, on the other hand, we have a budget of 20 for the experts, we can allocate more cases to them provided that the expected marginal gain from the expert exceeds the expected marginal gain from the rules. This directs more cases to the experts, and the objective function recovers somewhat, but it cannot exceed the expected as that is in itself an optimum as we have not changed the case-mix. The results are shown in Table 35.

	Expected	Actual	Optimal
Cases	900	900	900
Good	366	366	368
Bad	77	77	71
Cases referred to the expert	200	400	400
Cost	10	20	20
Objective	279	269	277

Table 35 Impact of More Experts

10.3 Changes in Relationships

We only have the one data set to work with, but it is possible, over time, that relationships between the attributes and outcomes change. Dealing with this presents quite a challenge:

- There can be a significant time between acceptance and outcome, meaning that any change in outcomes reflects the nature of the applicants, perhaps even years ago.
- The current framework (necessarily) side steps this problem by using outcome data (on old cases) to create rules for the current cases
- While we are estimating probabilities that cases will turn out to be good, or bad, we are not estimating when a case will turn bad. To make any sensible adjustment, we would have to wait until loans are repaid for any given cohort, in the case of Lending Club that can be 36 or 60 months.

As with transaction abandonment, we could model the relationship between the number (or proportion) of loans that go bad as a function of time and use that to inform the necessity to revisit the underlying models and rules. The training data would then consist of the most up-to-date outcomes (for loans that have run their course), with the oldest data being removed.

10.4 Summary

The framework we have created can deal with the situation where the number of experts is fixed and the caseload, or case-mix, changes. We solve this problem by increasing or reducing the number of cases that are processed by the experts in order of their marginal increased contribution to the profit function. If we have too many cases, we look for those with the lowest marginal contribution when they are directed to the experts. When we have too few, we look for the next highest.

11 FRAMEWORK

11.1 Introduction

Here, we present and describe one of the key research outputs, the framework and procedure to build a set of rules to optimise the credit assessment process and other similar processes that result in decisions. The framework is designed for the credit approval process but, based on the analysis in 5.2, it applies to a wide range of similar service business processes.

This framework includes:

- Computation, reasoning and allocation rules to determine how the cases are directed through the process with initial acceptance and rejection, requesting further information, with or without referral to the expert
- The determination of the number of experts required for any given caseload and case mix
- Operational optimisation, where we adjust the rules to make the best use of the available experts

A useful framework will use reliable methods that are widely accessible. The objective is to use methods that are practical, widely available and easy to use. The framework must also cope with practical difficulties, such as unequal sample sizes, discretisation, feature selection and missing data (such as outcome data for rejected cases). Finally, creating business rules in a familiar format, such as logical rules or weighted scores that reflect current business practices, are preferred over black-box methods that are difficult to interpret or understand how an answer was obtained.

First, we cover the assumptions and prerequisites for the framework. Then we describe the process of data preparation, modelling and estimating outcomes for the rejected cases. Finally, we identify how the rules should be built and applied.

11.2 Pre-Requisites

We assume that we have the following historical data available:

- A set of applicants (cases)
- For each case we have attribute values, such as credit rating, income
- A subset of cases that have been accepted
- Outcomes for accepted cases
- A subset of cases that have been rejected (with unknown outcomes)
- A means to estimate the potential profit from a good case, and the potential loss from a bad case
- The cost of processing each case (where applicable) by an expert
- Data around transaction abandonment to fit the Weibull distribution

We use historical data to fit models for the accuracy of the expert and the relationships between attributes, outcomes, and transaction abandonment. We can later combine that with forecast data to determine resource levels.

11.3 Data Preparation

The historical datasets we have can be viewed in Table 36:

Set	**Attributes**	**Judgement**	**Outcome**
Rejected (R)	Known	Reject	Unknown
Good cases (G)	Known	Accept	Good
Bad cases (B)	Known	Accept	Bad

Table 36 Data Sets

We have a data set of all the cases that are to be determined, and their attributes. Some of the cases are accepted and others are rejected in the normal course of business, and we know which these are. For those cases that are accepted, we have outcomes, but not for those cases that are rejected. We also know how many transactions were abandoned.

We have several tasks:

- Discretise, if necessary, the real attributes. This is required to overcome the inability of logistic regression to model non-linear behaviour

- Create a training/validation and test set to model the human expert. This consists of the cases that are accepted and rejected by the human expert
- Fill in the missing data for outcomes for rejected cases as we do not have outcome data for the rejected cases
- Create a training and validation set for the classification rules to determine the outcome. This consists of accepted and rejected cases, outcomes for the accepted cases and calculated outcomes for the rejected cases.
- Create a cost model for profit, loss and referral costs. Profit – from a good outcome – and loss – from a bad outcome – can be related to the attributes of the case—for example, loan amount. The referral cost model may include a fixed element and a cost related to the number of attributes.

11.3.1 Discretisation

We have determined that logistic regression is a good method to model the behaviour of the expert and the probability that the expert is correct. However, if there are non-linearities in the relationships, we need to discretise any real variables within the attributes. To keep this simple and practical, we can use unsupervised methods such as Equal Width (of the bin) or Equal Frequency (of elements in the bin). Supervised methods are more complex, but Entropy Based Discretisation produces good results and is easy to implement (Mark et al., 2011).

11.3.2 Training Set for Judgements

We need to create a model of the human expert to determine $P(Eg)$, the probability that the expert will judge a case to be good. For this, we only need the accepted cases (G and B in Table 36) and the associated judgements. If we are using cross-validation, we need a training/validation set and a test set. A typical split is between 70:30 and 90:10. The main consideration is an imbalance. If the proportion of either class is less than 10% in either case, we should consider bagging or boosting to better balance the data set.

We also need a training/validation and test set to estimate $p(good|Eg)$ which is the probability that a case is good when the expert says so. For this, we use the accepted cases and their outcomes. Similar considerations on split and imbalance apply.

The two quantities, $p(good|EG)$ and $p(EG)$ allow us to calculate the probability that the expert judges the case to be good, and that it is, indeed good:

$p(good\ and\ Eg) = p(good|Eg)p(Eg)$

And

$p(bad\ and\ Eg) = \left(1 - p(good|Eg)\right)p(Eg)$

11.3.3 Initial Feature Selection

At this stage, it is worth eliminating any attributes that do not add value. We do not yet have the full objective function, but we can use a filter method to remove attributes that are closely correlated or redundant. This can be done using the training set of attributes and judgments before we fit any models of the human expert. Note that there is a potential element of iteration in this process as to fill in the missing data for outcome of rejected cases we use the models that we build using the attributes that we select. So, we cannot use outcome data for feature selection at the beginning. Note that we can only assess attributes for which are included in the data set, and hence our only option is removal. Attributes can, of course, be added, data collected and then assessed, and that would be possible within this framework.

11.3.4 Missing Data

Unfortunately, we do not have outcome data for rejected cases. Still, we can use the method in 6.4 to estimate the probability that a rejected case is good or bad and use that probability to classify the case as good or bad.

We use the confusion matrix where the elements a, b, c and d are the relevant probabilities. For each case in we have $p(Eg)$ and for the accepted cases we have $p(good|Eg)$. To recap, Table 37 gives the confusion matrix.

	Judged good	**Judged Bad**
Good	a	b
Bad	c	d

Table 37 Confusion Matrix for Missing Data

We have:

$$a + c = p(Eg)$$

$$\frac{a}{a + c} = p(good|Eg)$$

$$\Rightarrow a = p(Eg).p(good|Eg)$$

To ensure $c \geq 0$ (in case we have some noise in the calculation of $p(Eg)$) we set

$$c = \max\left((p(Eg) - a), 0\right)$$

As in 6.4 then calculate the other variables assuming:

$$\frac{a}{(a + c)} = \frac{b}{(b + d)}$$

We would expect that the proportion of unknown cases that are good to be similar to the proportion of accepted cases that are bad. If not, we should check our calculation or look for some explanation.

11.3.5 Cost Models

There are 3 cost models for this and similar service business processes to which it can apply.

11.3.5.1 Profit and Loss

The potential profit is the upside; how much money can we expect to make if things go well and our judgement is correct. The loss function is the opposite; we lose money if our judgement is to accept, and then, it proves to be incorrect. This can only really be determined on an application by application basis. For credit approval, we need data on issues such as:

- Profit: gross interest, financing costs and expense ratios
- Loss: mean time to default, net debt recovery and recovery costs

11.3.5.2 Referral Costs

We can assume that this consists of two elements:

- Fixed familiarisation cost (the same for each case)
- Additional (on a per attribute basis)

Referral can also be expressed relative to the size of the loan if this is more representative. For the initial rules optimisation problem, it makes sense to use referral cost, as the objective is to optimise profit. When we move to operational optimisation, it may be better to use the time taken and then compare that to the time available for referral based on the number of experts available.

For the sake of simplicity (and in the absence of any data), we can assume that each attribute takes the same time and we have:

$t(k) = (t_I + kt_A)$

Where

t = total referral time

t_I = initialisation of the process

t_A = time to assess each attribute

h= hourly cost of an expert

11.3.5.3 Transaction Abandonment

The relationship between the number of attributes requested and the proportion of transactions abandoned has been identified in 8.3. For any new problem, we need to fit the values of β and λ. This can be done using the Weibull plot where we plot $\ln(-\ln(1-F))$ (vertical access) against $\ln t$ where F is the proportion that abandon and t is the time. The straight line indicates a good fit, the slope of the line is equal to β, and the intercept is $-\ln \beta\lambda$.

We then have the proportion of transactions abandoned as a function of time given by:

$$F(t) = 1 - \exp - \left(\frac{t}{\lambda}\right)^{\beta}$$

Where

$t(k) = t_I + kt_A$

Where

t = total transaction time

t_I = initialisation of the process

t_A = time to respond to each request for information (assumed to be the same for each)

k = number of attributes

11.4 Business Rule Creation

11.4.1 Referral

To decide whether the rules or the expert determines a case, we look at:

$$Egain = p(Eg\ and\ good).profit - p(Eg\ and\ bad).loss - cost$$

$$Rgain = p(Rg\ and\ good).profit - p(Rg\ and\ bad).loss$$

And if

$$Egain - Rgain < cost$$

We allocate the case to the expert. Otherwise, it is allocated to the rules.

The decision process is given to the rules or the expert depending on the direction above. We incorporate this logic into the rules in the next section.

11.4.2 Classification

The data to create the rules consists of rejected (R), good (G) and bad (B) case data. In the case study, we assumed that misclassification costs were equal, but in general, this will not be the case. To overcome this problem, we need to create a training/validation set that has the right proportion of good and bad cases to match the unequal misclassification costs. For example, if the profit from a good case is twice the loss from a bad case, we need twice as many bad cases in the training/validation set to compensate. We can do this with bagging (removing excess data) or boosting (creating additional data). The choice of rule building (decision trees, regression, etc.) is determined by performance (on the given data set) and the desired simplicity of the final rules.

We also require the performance of the rules in terms of $p(good\ and\ Rg)$. This can be determined directly from $p(good)$ should we use logistic regression or by inspection of each leaf of the decision tree when it is applied to the test set. Similarly, we can estimate $p(bad\ and\ Rg)$ from the number of bad cases in the leaves of the decision tree.

11.4.3 Combined Business Rules

The referral logic and classification are combined to create the business rules using the expected net gain given the attributes we have (which can be all of them or a selected subset) of the rule decision or expert assessment. This creates the rule sets that are necessary to carry out feature selection properly.

11.5 Feature Selection

This is an important step as redundant attributes are expensive as they increase transaction abandonment.

As identified in Chapter 7, the wrapper method requires the complete set of rules and costs to be incorporated to determine the objective function. For reasons of efficiency, it is worth setting up a process where attributes can be added or removed that drives the steps above.

Note that we have already created a model of the expert and classified the rejected cases into good and bad. We do not need to do that again.

Once we have selected a subset of attributes, we carry out these steps:

- Calculate referral costs, profit and loss, and transaction abandonment as in 13.3 for the chosen attributes
- Modify the expert model as per section 11.4. This re-estimates the accuracy of the expert based on a reduced set of attributes.
- Calculate $p(good\ and\ EG), p(bad\ and\ Eg)$. This is the new model of the expert.
- Re-train the classification rules using the new attributes.
- Calculate $p(good\ and\ RG)\ and\ \ p(bad\ and\ Rg)$. If we use a decision tree or rule learning, these are obtained from the contents of the leaves on application to the test set. For logistic regression, we calculate probabilities directly, and we use $p(good)\ and\ p(bad)$
- Calculate $Rgain, Egain$ as 13.4.1. This determines the accept, reject or refer rules for each case.
- Evaluate the new rules on the test set
- Calculate net profit

11.6 Interactive Integration

11.6.1 Introduction

This process is somewhat different from a typical classification process as we effectively have three options after processing a subset of attributes:

- Decide
- Refer
- Request more information

The attributes can be requested in blocks – as in an initial screening process – or one-by-one, in which case the decision to request also specifies the next attribute.

There are two options, as identified previously:

- Logistic regression to mode the expert and performing the classification
- Logistic regression to model the expert and a decision tree for classification

The key difference is that logistic regression calculates $Rgain$ directly on a case by case basis. If we use a decision tree, we apply the tree to the test set and determine $Rgain$ from the contents of each leaf. The following sections apply to both but we can exploit the property of a decision tree that effectively determines the best order to request attributes and terminate when nothing further can be gained rather than apply the process in 13.6.3.

11.6.2 Blocks

Interactive integration consists of finding a block of attributes that are initially requested and then having the option to decide (using the rules), request for further information or refer to the expert. To create an interactive integration, we must:

i. Rank the attributes in descending importance (in terms of their contribution to the overall objective function)
ii. Decide how many attributes to request initially.

Let us suppose that we have the k highest ranked attributes. Using our model, we can then calculate $Rgain(k)$ and $Egain(k)$ for any case and any value of k. Then we apply the following algorithm for each value of k. Note that when the attributes are requested, we must allow for the potential to ask for more information (the remaining attributes) or refer to the expert, or both.

With n cases and m attributes, the objective (as a function of k) is given by the following expression

$$\sum_{i=1}^{n} \max\left(Rgain_i(k), Egain_i(k), Rgain_i(m), Egain_i(m)\right)$$

Where

$Rgain_i(k)$ is the expected gain using k attributes on case i using the rules (decide now)

$Egain_i(k)$ is the expected gain using k attributes on case i using the expert (refer).

$Rgain_i(k)$ is the expected gain using k attributes on case i using the rules (decide now)

$Egain_i(k)$ is the expected gain using k attributes on case i using the expert (refer).

$Rgain_i(m)$ is the expected gain using k attributes on case i using the rules (decide with more information)

$Egain_i(m)$ is the expected gain using k attributes on case i using the expert (refer and then decide with more information)

11.6.3 Individual

The attributes can also be ordered and processed individually. In this instance, we can use concepts from feature selection to identify the order, taking into account subsequent options for a decision, referral or additional information. Dropping the suffix i for clarity we have:

$$Gain(1) = \max_{j}(Rgain(j), Egain(j)) = \max(Rgain(j_1), Egain(j_1))$$

And then

$$Gain(2) = \max_{j \neq j_1}\left(Rgain(j), Egain(j)\right)$$

$$Gain(m) = \max_{j \neq (j_1, j_2, j_{m-1})}\left(Rgain(j), Egain(j)\right)$$

Then, to identify the optimal point to decide or refer, we choose the highest value. Note that we are still working out the optimal referral scheme and, as such, optimising the number of experts required for a given caseload.

11.7 Operational Optimisation

This problem differs to the preceding ones in three critical ways:

i. The number of experts has been determined and is fixed

ii. The cases are arriving in real-time, and the caseload and case-mix is unknown (although we can expect them to resemble the training set)

iii. The objective is to maximise profit and minimise (or avoid) queueing

Previously we have used the net gain functions $(Egain(k) and\ Rgain(k))$ as the basis of our decision making and optimisation. We now form different quantities $(Rmgain(k)\ and\ Emgain(k))$ that represent the marginal gain from processing cases using k attributes.

$$Emgain(k) = (p(Eg\ and\ good).profit - p(Eg\ and\ bad).loss)(k)$$

$$Rmgain(k) = Rgain(k)$$
$$= (p(Rg\ and\ good).profit - p(Rg\ and\ bad).loss)(k)$$

The reasoning is this: expert numbers are fixed, and the cost is incurred regardless of the number of cases referred. As such, the cost is independent and can be ignored. We are therefore interested in the cases that will deliver the most (gross) profit, which is represented by the $Emgain$ and $Rmgain$ functions.

At the simplest level, where we request and process all the attributes, we have

$$Mgain_i = \max\,(Rmgain_i, Emgain_i)$$

$$Refer_i = \ t(m)\ if\ Emgain > Rmgain$$

Where $t(m)$ is the time required to process a case with m attributes.

We order $Mgain$ in descending order and stop when $\sum Refer_i = T$ where T is the total time available to the experts. In this way, we utilise the time available to process the cases that deliver the greatest gain.

We can extend this idea to any level of complexity (blocked attributes or individual case optimisation) by

- substituting $Emgain$ for $Egain$
- Calculate $Mgain_i$ for each case
- take the cases with the largest value of $Mgain$ subject to $\sum_{i=1}^{n} t_i \leq T$

11.8 Application

We now apply this framework to a different, but similar problem: that of credit card approval. We use the Australian Credit Approval Database (Dua et al., 2019) that contains 690 records of successful and unsuccessful credit card applications. In common with many other such datasets, this only contains one of either judgements or outcomes (in this case judgements). To make this usable, we introduce random differences between judgements and outcomes so that about 15% of successful applicants are deemed to be bad customers, and 15% of unsuccessful applicants are deemed to be good customers, turned down in error. This is consistent with the Lending Club experience, but in practice, any percentage could be used. The Auscredit database is not labelled – for reasons of confidentiality - and is scaled from -1 to +1, as shown in Table 38.

Attributes	**Min**	**Max**	**Average**
X1	-1	1	0.3565
X2	-1	1	-0.4641
X3	-1	1	-0.6601
X4	-1	1	-0.2333
X5	-1	1	-0.01962
X6	-1	1	-0.07681
X7	-1	1	-0.08406
X8	-1	1	0.04638
X9	-1	1	-0.1440
X10	-1	1	-0.9284
X11	-1	1	-0.08406
X12	-1	1	-0.07101
X13	-1	1	-0.8160
X14	-1	1	-0.9797

Table 38 Auscredit Database

11.8.1 Assumptions

In the absence of data on classification and misclassification costs, we assume that the benefit of a good case and the cost of a bad case are both unity, and the cost of classification (by the expert) is 0.05 (5%). This is consistent with the analysis of the Lending Club data.

11.8.2 Modelling & Rule Building

The dataset has only 690 elements, so we use 10-fold cross-validation for training and validation purposes. We also take a sample (using the sampling without replacement function in Weka) of 1/3 to test and evaluate the rules that we build.

Experiments on the training set with Weka show that decision trees outperform rule building and logistic regression. Classification accuracy for decision trees is between 70 and 80%, with the other methods between 60 and 70%. The best method is random forest at 77%, but this does not produce a decision tree resembling typical business rules, so we choose the J48 tree (Quinlan, 1993). In this case, we have ensured that the classification accuracy of the expert is 85% (because of the 15% error rate randomly introduced), so we expect at least some of the cases would be directed towards the expert.

For modelling the instance by instance performance of the expert and the rules, we use logistic regression. The creates four models:

- P(good and Eg) probability that the expert accepts and the case is good
- P(bad and Eg) probability that the expert accepts and the case is bad
- P(good and Rg) probability that the rules accept and the case is good
- P(bad and Rg) probability that the rules accept and the case is bad

Using these quantities, we can express Egain and Rgain for each instance and then decide on the best way to classify each instance. The final step is to integrate the rules and the expert. For this we could fit a decision tree – the final set of rules – based on the Accept, Reject or Refer status. In this set of data, the classification error is nearer 50% with poor performance on the good cases. The alternative is to use logistic regression as a classifier, but we have already noted poor performance, so the pragmatic approach is to use the J48 tree and determined referral on a node-by-node basis. This does have the potential disadvantage that the calculation of the 'goodness' of a node may be inaccurate. Still, we can err on the safe side and refer if in doubt. Another advantage of using the decision tree in this way is that we can see how many data items are requested and create a similar map of the number of items required and when the decision is made. Figure 22 shows a portion of the tree and the ACCEPT, REJECT or REFER decisions. It also shows the ability of a decision tree to reach a conclusion with only part of the data.

At the end of each leaf, we calculate the merits of:

- Accept, where we get the net gain (all cases in the leaf whether they are good or bad).
- Reject, where we nothing.

- Refer, where we get the expected benefit of expert assessment.

This method is preferable to identifying accept, reject and refer, and the reclassifying as classification accuracy for the 3-class problem is much worse than for the 2-class problem. Also, fixing the tree structure makes it possible to adjust the costs without refitting the tree.

J48 pruned tree

X4 <= -1

| X1 <= -1

| | X7 <= -0.947368

| | | X8 <= -1: Y (4.0) **ACCEPT after 4 data items**

| | | X8 > -1

| | | | X2 <= -0.719399: Y (2.0) **ACCEPT after 6 data items**

| | | | X2 > -0.719399: N (2.0) **REJECT after 6 data items**

| | X7 > -0.947368: N (5.0) **REJECT after 3 data items**

| X1 > -1: N (45.0/11.0) **REFER after 2 data items**

X4 > -1

| X8 <= -1

| | X6 <= -0.75

| | | X11 <= -1

| | | | X2 <= -0.142857: N (2.0) **REJECT after 5 data items**

| | | | X2 > -0.142857: Y (3.0) **ACCEPT after 5 data items**

| | | X11 > -1: Y (2.0) **ACCEPT after 4 data items**

| | X6 > -0.75

| | | X1 <= -1

| | | | X3 <= -0.166786

| | | | | X11 <= -1

| | | | | | X2 <= -0.451128: Y (6.0/1.0) **REFER after 7 data items**

Figure 22 Using Tree to Determine Referral

This method removes the key disadvantages of decision trees and enables one of the critical advantages of context-sensitive decision making using only the minimum number of attributes. While this can be done with logistic regression, it requires more steps and a degree of complexity.

11.8.3 Results

The results in Table 39 show an even split between the rules and the expert. Just over 50% compared to the previous results of 80% and 70%.

VALUE	**GOOD**	**BAD**	**COST**	**NET**
ORIG	91	14	11.5	65.5
NEW	102	23	5.6	73.4
CASES	EXPERT	RULES		
ORIG	230	0		
NEW	112	118		

Table 39 Benefits of Referral

The ORIG line is with the expert processing all the cases and the NEW line is with the expert and the rules. GOOD refers to the value of good cases accepted and BAD refers to the cost of bad cases accepted COST is the referral cost (which for the rules is zero). NET is GOOD-BAD-COST, the overall benefit to the organisation.

This shows that there is a better net result of 73.4 if we combine the rules with an expert. Previously the expert achieved a net profit of 65.5 by accepting 91 good cases and 14 bad cases, and with a processing cost of 111.5. The combined result reduces processing cost to 5.6, accepting 102 good cases and 23 bad cases.

11.8.4 Feature Selection

Unlike Lending Club, Auscredit has enough attributes (14) to carry out feature selection. Using the wrapper method in Weka (Kohavi et al., 1997) we get:

Selected attributes: 1,2,3,4,6,7,8,10,11,12 : 10

We can refit the models and rebuild the rules using these attributes. The results are similar, with more cases going to the rules. This can be explained since the feature selection uses the J48 tree and as such, avoids degradation. However, with the expert, we model the reduction in Egain by taking terms out of the logistic regression and rebasing the function.

We can estimate the potential gain of reducing the attributes from 14 to 10 using our model of transaction abandonment. With ten attributes, we get another 8% of potential customers completing the process. All other things being equal, that gives an increase in the net benefit of 6.3 to 72.8. However, further reduction in the number of attributes reduces the classification accuracy by more than the increase in customer retention. Table 40 gives the results.

VALUE	**GOOD**	**BAD**	**COST**	**NET**
ORIG	97	15	12	70
NEW	107	24	5.5	77.5
CASES	EXPERT	RULES		
ORIG	243	0		
NEW	110	133		

Table 40 Results with 10 Attributes

We can see the impact of transaction abandonment for attributes 10 to 14 in Table 41, below.

Data Item	8	9	10	11	12	13	14
Time	24	27	30	33	36	39	42
Remain	0.73	0.72	0.70	0.69	0.68	0.67	0.65

Table 41 Abandonment 10-14 Attributes

11.8.5 Rule Adjustment

The final aspect of the framework is the ability to adjust the rules should the caseload (or case-mix) change. In the test set (above), we had 104 referrals for 230 cases which is a referral rate of 45%.

If we take a random sample of 200 cases, we see that the optimal referral rate – for these cases – is 51.5% on the assumption that referral costs 0.05 per case. We can optimise and reduce this back to 45%, and we get the results in Table 42

VALUE	GOOD	BAD	COST	NET
ORIG	86	20	5.15	60.85
NEW	84	26	4.5	53.5
CASES	EXPERT	RULES		
ORIG	103	97		
NEW	90	110		

Table 42 Impact of Reducing Referral Rates

If, on the other hand, we could make more experts available, for example, enough to refer 60% of the case we would have the results in Table 43:

VALUE	GOOD	BAD	COST	NET
ORIG	86	20	5.15	60.85
NEW	88	19	6.00	63.00
CASES	EXPERT	RULES		
ORIG	103	97		
NEW	120	80		

Table 43 Impact of Increasing Number of Experts

11.9Results

The application of the framework has demonstrated that better result can be obtained by using rules and experts together. This is due to some critical factors:

i. Rules can be trained on outcomes that are different from expert judgments. They can also be retrained when circumstances change.

ii. There is a cost, generally per transaction, of expert assessments and this cost can outweigh the benefit of referral

iii. When rules are trained on outcomes, their performance is comparable to the expert and can be used on the majority of cases

The overall benefit of a few percentage points increase in revenue is significant in terms of profitability.

The results also show the impact of transaction abandonment and the importance of curtailing the application process as soon as possible, potentially in a context sensitive manner where the next question depends on the response to the last. This is different from the typical approach to feature selection where an attribute is either present or not present (and incurs cost or does not). In our situation, attributes are available but not necessarily utilised, and only incur a cost if they are used.

We have also observed the importance of on-line adjustment. If the caseload changes (or the number of experts) changes then the rules must change to send the right number of cases for referral. Too many cases can create a bottleneck, and two few means that valuable resources are idle.

In terms of method, we have used logistic regression to model the expert and either logistic regression or decision trees to create the rules. Logistic regression has the advantage of calculating probabilities directly and is preferable for modelling the expert. For the rules themselves, both methods can create a context-sensitive set (where the question depends on the answer to the last) that terminate when there is no benefit in asking for further information. We anticipated a disadvantage with decision trees with the requirement to rebuild should misclassification or referral costs change. However, we have found a pragmatic solution to the problem by classifying each leaf in a binary tree rather than having three classes within the tree.

The method that we created to estimate the outcome for cases that were not accepted cannot be tested against any data. However, the results are reasonable as its application resulted in a similar percentage of (estimated) false negatives as there were false positives judgements.

The are other considerations that can be included, such as differences in expert costs and performance, more complex misclassification and referral cost models. The same considerations apply to transaction abandonment; with more data, more elaborate models can be built and tested.

11.10 Application to other Service Business Processes

In Chapter 4, we identified service business processes with similar characteristics to credit approval. In each case, we want to make a binary decision such as accept/reject, proceed/do not proceed, fraudulent/honest, problem/no problem, etc. The basis of this determination is what is known, or can be discovered about the situation, and rules or human experts can make the decision.

These include:

- Loan/insurance application

  The objective is to maximise the number of applicants we accept and minimise the amount that default. The application process consists of providing information in a set format, with follow up documentation.

- Insurance claim

  We need to determine which claims are valid (there is a policy in force, and the type of claim is covered) and not fraudulent. Information is derived from the case and the characteristics of the claimant. The objective is to minimise any fraudulent claims that are accepted – in error – and avoid customer dissatisfaction when honest claims are referred and questioned.

- Making an investment

  The rules around investment are generally one of compliance (are we allowed to invest in this asset as a policy) and the CRA rules act on the anticipated return and risk information.

- Entering into a contract

These rules are like those for investment, and the considerations are ones of capability, expected return and risk. There is typically a set of rules based on contract value and the seniority of the final decision maker.

- Deciding on a merger or acquisition

  Rules around mergers and acquisitions consider expected returns (from synergies, efficiencies and market position), costs (rationalisation and change), and risks in terms of cost and time overruns or failure to achieve the expected returns. Like contracts, there will be limits on the level of deal that staff can sign off on.

- Staff recruitment

  Recruitment is like credit approval. Each applicant provides information that can be tested against criteria such as qualifications, years of experience, previous salary, etc. One of the main objectives – like credit approval – is to avoid a mistake that later becomes apparent.

- Diagnosis

  Simple diagnosis, such as whether a patient has a condition or not, fits within this list of binary decisions. Triage is the same and fits into our model well. In both cases, we have measurements or responses that can trigger a set of rules. Identifying condition – from a list – is an integer problem and more complex.

- Fault finding

  Fault finding can be binary, but more often integer. It is the same as a diagnosis.

- Sales

  Like diagnosis and fault-finding, the sales problem can be integer (what to offer the customer) or binary (should we make the offer).

- Child protection

  The binary decision for child protection is in the triage process; urgent or not. Further on, there is a question of what sort of intervention, if any. Like insurance claims, there is data on the case and the participants.

- Fraud detection

  Fraud detection is binary, and the rules use information like credit application. The amount of money involved, what was purchased, on-line or in person, etc.

- Visa application

  This process is like fraud detection, and we are attempting to identify fraudulent applications in both substance (the facts are wrong) and intent (the applicant intends to use the visa for another purpose.

11.11 Summary

Here we have presented a process to build a set of rules based on a familiar situation:

i. A defined objective function for the business
ii. A set of cases with attributes that are useful for making a decision
iii. A set of accepted cases with outcomes
iv. A set of rejected cases without outcomes
v. Data on transaction abandonment

From these data, we have developed a method to build a set of rules that:

i. Processes the cases, asking for attributes in the most effective order
ii. Deciding, at each point, whether to accept, reject, continue or refer to the human expert
iii. Can adjust when the caseload or case-mix diverges from that expected

The framework is designed to be robust and easy for a non-expert to apply using tools such as Weka and tried and tested methods. Based on our previous analysis, the framework is applicable to a range of service business processes that share the characteristics of credit approval. These include processing insurance claims, investment appraisal, recruitment, and medical diagnosis.

12 CONCLUSIONS

12.1 Introduction

Business rules are pervasive within the services sector. They provide consistency and allow relatively unskilled staff to process complex transactions correctly. But the rules may also have an impact on the costs and profits of an organisation. Financial services, transport and human services are areas where the rules themselves can predictably impact the bottom line. This thesis has identified and applied methods from machine learning and mathematical optimisation that can build a set of rules that will maximise profit, performance or customer service, or any other key performance indicators. The manufacturing, energy and process industries have embraced mathematical optimisation techniques to improve efficiency, increase productivity and reduce costs. The different nature of services, where customers are different and require individual attention, makes business rule optimisation attractive, beneficial and because of this research, practical.

There is recognition in previous research that business rules impact the performance of an organisation. What is missing is the link between the business rule choices and the profitability of the organisation, and the potential for rules to augment instead of just replacing the human decision maker. Addressing this has enabled us to frame and solve a business rule optimisation problem.

As business rules have been applied to a wide range of processes, this research has focussed on a specific class (CRA rules) and developed theory around framing the optimisation problem, the nature of the objective function and methods to find the optimal set of rules. The research applied the theory to a case study with the necessary data set. In this way, it tested the concepts and demonstrated that there is potential value. A practical framework was also developed that could be applied in any business process where decisions need to be made, and where there is potential merit in combining automated business rules with human experts.

This thesis has considered the broader question and then focussed on a set of reasonably common problems where the objective is, ultimately, to make a binary decision such as accept or reject, approve or not, or proceed, or not, for example. Limitations were placed on the methods we use to build the rules, the form of the rules, and the ability to understand how the rules reach a decision. The concepts were then applied and proven in two representative case study examples. The research question and contributions should, therefore, be read in this context.

12.2 Research Questions

The overarching research question concerns building rules that provide the best expected outcome for an organisation.

How can business rules and the process of their application be designed for better business results?

The thesis investigated whether business rules and the process of their application could be designed for better business results for a set of service business processes that have the same characteristics as credit approval. The theory was tested using real data for two completely different examples of credit approval utilising the case study approach. As such, we have created a methodology and framework that is applicable to the family of rules that address classification problems and shown that they can deliver better business results using a representative case study and different data sets.

12.2.1 RQ1

How is the optimisation problem to be defined?

The problem is to choose that set of rules, to act upon information that we must maximise the expected profit over an anticipated caseload. The optimisation variables are the choice and characteristics of the rules. The objective function is the expected profit over a set of historical cases, or anticipated cases. The output includes the rules themselves and economic consequences in terms of profit (from a good decision), a loss (from a bad decision) and the cost of collecting and processing information. Other business rules are proscribed, such as legal or statutory rules, but these are not constraints in the rule building optimisation problem, but of course, they are constraints in operation.

12.2.2 RQ2

How should the rules be built and optimised?

We deliberately limited our choice of rules to standard industry formats: IF-THEN-ELSE and weighted sums, but with the explicit option to refer to a human expert should that be the optimal choice, based on all the relevant cost factors. With this limitation, the research has identified two practical and reliable methods to build such rules: decision trees and logistic regression. The choice of method is problem-dependent; how well the classifiers work and whether there are non-linearities and dependent variables, for example. Problems can also be addressed using a hybrid method consisting of decision trees and logistic regression. Rule building can be found in Chapter 7.

12.2.3 RQ3

What information should be requested initially and subsequently?

Models have been created of the costs and risks associated with obtaining information and created a method to balance the benefits, the costs and risks of obtaining more information and deciding whether the decision should be made by the rules or the expert.

The information required can be determined as part of the design a decision can be made whether to obtain more within the rule structure.

12.2.4 RQ4

How to incorporate data on judgements (decisions) and outcomes?

The essential idea is one of conditional probability and the set of accepted cases, and their outcomes are used to determine the probability that a case is good (or bad) when the expert has judged it to be good. Data on accepted and rejected cases is used to determine the probability that an expert will judge a case to be good (or bad). This allows the calculation of the expected gain of an expert making the decision on any given case. Lending Club data on outcomes have been used to build the rules and determine the probability that a case will be a good outcome.

12.2.5 RQ5

How to incorporate a human decision maker (expert) to best effect?

The option of referral has been included within the rules and uses the expert if the net benefit is greater than using the rules. This is covered in Chapter 7 and then developed in Chapters 10-12. The analysis has assumed that communication with a customer consists of requests for specific information. It has not, for example, admitted the explicit potential for an expert in communicating directly (by telephone or email for example). Still, if that occurs in practice, communication will impact the expert's judgement accuracy, and this will could be captured and modelled.

12.2.6 RQ6

On a case by case basis, how to decide between the outcome of the rules and the potential to refer to the human?

Logistic regression has been employed to model the decision of the expert and the outcomes. This calculates the probabilities that the rules and expert decide that the case is good and the conditional probabilities that the case will be good GIVEN the rules or the expert so do. This enables us to calculate the expected benefits (net of cost) of the two options. Probabilities and costs can be calculated on a case by case basis. This is explained in Chapter 8.

12.2.7 RQ7

How can the rules be adapted to maintain maximum efficiency in changing circumstances?

The off-line optimisation problem – including information, rule building and interaction with experts - has no constraints one of the outcomes is the optimal number of experts required. When the caseload or case-mix varies, there is a new constraint (the number of capacity of the experts) but the marginal cost of referral is zero because they are employed (and paid) regardless of the actual number of cases that should be referred. Similarly, we could decide to work them overtime by weighing the cost of overtime versus the benefit. This problem has been solved by taking the rules in order of their gross contribution. This is covered in Chapter 10. Note that the research has not addressed adaption to long-run trends in customer behaviour. This would include detection – that the model is degrading - as well as adaption. Detection would require some form of goodness-of-fit or significance testing.

12.3 Contribution

The contributions of this thesis to the field of business rules, and the application of machine learning and mathematical programming are five-fold. They address the problems inherent in the construction of business rules that optimise expected net profit for a specific class of service business processes that result in a decision (as identified in 4.2). The contribution includes methods to build the rules, integration of business rules with human decision makers, a model of transaction abandonment and its integration with the rules, a proof of concept using real data on judgements, outcomes and transaction abandonment, and a framework for implementation.

12.3.1 RC1

An analysis of the characteristics of rule building methods and identification of the strengths and weaknesses from the perspectives of reliability, complexity, flexibility and utility.

> The following methods have been analysed in Chapter 7: first principle, rule learning, decision trees and (logistic) regression. First principles method is complex and potentially unreliable. The other three have advantages and disadvantages, depending on the nature of the problem. A hybrid approach – using logistic regression to model the human expert – and a modified decision tree to execute the rules has advantages of performance and elegance.

12.3.2 RC2

An extended version of the LENS model (Brunswik, 1985) using logistic regression with outcomes, the expert decision and, now, additionally the rules decision.

> The original LENS model is two dimensional and based on linear regression. This research has created an extended version using logistic regression that is better suited to binary decisions than linear regression. Logistic regression deals better with non-linearities and provides a case by case estimate of the probability that a case is good or bad, and the probability that the conclusion

of the expert judgement and/or the rules are indeed correct. This is explained further in Chapter 7.

12.3.3 RC3

Incorporating a model of transaction abandonment using the Weibull distribution (Weibull, 1951).

Chapter 6 has shown that the Weibull distribution is a good fit for openly available collected data in two domains around transaction abandonment. The estimated parameters of the distribution accord with the infant mortality phase. this would be expected from a large data set which is essentially a survival process. The publicly available data only consists of time and transaction abandonment and does not enable any analysis of the type of question on it. There is, however, a link between the amount of information requested and the time taken. Determining the optimal amount of information in this context is an application of machine learning with feature selection where the cost of information increases as the process unfolds. If we had more specific data, question by question, we could modify our approach and assign each question a time weighted value or abandonment factor. This is a potential area for future research.

12.3.4 RC4

A proof of concept that optimises business rules in the wider context and demonstrates that this is feasible and useful on an example with real data

Using real data from Lending Club, we have optimised rules for the initial selection, final selection and operation phases. This is covered further in Chapter 8 for the design of the static system and Chapter 9 for the interactive system. We also describe the operational implementation in Chapter 10. The business rules are optimised considering the accuracy of the human expert's decision on any particular case, the cost of human intervention, the potential for transaction abandonment as we ask for more data, and the relative merits of the rules making the decision, or deciding to refer to the expert. The resultant rules have been tested on a validation set and found to be better than the human expert or completely automated system.

12.3.5 RC5

A framework and guide for building rules that optimises the general problem of customer selection and acceptance

This framework and guidelines are the realisations of a novel, and mission directed research exercise (including elements of machine learning, modelling, logistic regression and probability) that will enable users to build their own application using easily accessible tools and minimal data manipulation. This is given in Chapter 11.

12.4 Research Hypotheses

12.4.1 H1

The basic hypothesis is that it is possible to optimise business rules in the sense that they give the best results over a defined range of situations (either determined by analysis of historical data or forecasts) considering:

- Data on outcomes and judgements
- The potential to refer decisions to experts
- The costs of asking for further information
- The impact of further information

Chapters 8 and 9 validate this hypothesis for the case study, which is representative of many common service business processes and includes all the elements above. We can conclude that – subject to limitations on the types of rules - an optimal (but not necessarily unique) set will exist for CRA problems based on the theory developed in Chapter 5. If this were not true, the results would always be the same for each potential set of rules, which themselves would provide (non-unique) optimal results. However, there is no guarantee that we will always find the optimum given the nature of mixed-integer optimisation problems.

12.4.2 H2

A secondary hypothesis is that business rules can be adapted to take account of circumstances that differ from those already experienced or anticipated.

Chapter 10 developed and proved a method to adapt the rules to account for a change in caseload or case-mix using the data from the case study. It concludes that that (by definition) CRA rules have an impact on resource requirements. This is a new optimisation problem with the same objective but with an additional constraint which is that the resources are equal to those that are available.

12.5 Limitations and Future Research

12.5.1 Limitations

Business rule optimisation is a new area of research and has the potential to develop into a completely new field. The research in this thesis has been – necessarily – confined to class of problems in the service sector that require a decision to be made based on data available where we have the opportunity to use a set of rules – evaluated automatically – and a set of experts. The rules have several options, decide positively, decide negatively, refer or request more information. The final decision of the process, however, is binary; positive or negative. Based on processing mapping, we have shown that many processes in the service sector share the same characteristics in terms of the type of decision, the cost of gathering and processing information and the option to decide or refer to a human expert. These include processes within the financial, health and social services sectors.

We envisage a variety of ways in which this research can be extended:

- In the interests of clarity, simplicity, and practicality, we have limited the form of the rules to that which is familiar to the services sector and classification techniques that are widely accessible in standard packages. There are much more complex rule sets and classifiers. So, if anything, better results will be obtained with more sophisticated methods.
- Not all business rules resolve to a binary decision; there can be more than one choice. This quickly gets complicated; for example, with only three choices, we need potentially sic different values for misclassification costs. The methods we have employed (logistic regression and decision

trees) are not limited to two classes, and the probability that the decision is correct would have to reflect that. For example, calculate the probability of disease A, B or C being present when the doctor diagnoses disease A, etc. Also, within the process there is a range of options including decide, refer and request more information.

- While we do not have access to relevant data, human experts will differ and there is the potential to build rules that recognise this and direct cases to the expert that will add the most value and, in the operational sense, the expert who is also available or working on other cases that add less value
- Not all service business processes are about making decisions. Rules may guide how to proceed or how to do something. Provided that we can quantify the financial or customer service impacts of different guidance, these sorts of rules should be amenable to optimisation.
- Different forms of rules may be used in other sectors. For example, in roadside assist or transport, we may need to calculate driving distances and travel times to make decisions. In this case, different optimisation techniques will be required, such as non-linear mixed-integer optimisation, constraint programming or evolutionary computing.
- The real, unaddressed, problem is this: given a business problem and data, how do we build a set of rules that could conceivably be any function defined on the attributes or other situational information. This problem is like optimal control (Kirk, 2004) that addresses the problem of finding functions that control dynamic systems to minimise cost and maximise returns. There are similarities to BRO, and optimal control also (necessarily) limits the functions to certain forms, such as linear quadratic functions. Similarly, in this thesis the rule functions have been limited to decision trees and weighted sums and the problem is to identify the optimal parameters and coefficients.

12.5.2 Future Work

Business rule optimisation is a potentially large area for research. In this research, we have only explored CRA and operational rules in a generic services business problem with a binary decision; accept or reject. Also, due to availability of datit has focussed on the customer selection example, with Lending Club as a source of data. There will be other data that, while not currently openly available, that could be used to populate and further develop the framework for other CRA rule problems. This includes problems where the decisions are integer and real in nature, as well as binary.

There is also scope to apply other methods for rule building, such as evolutionary computing, that are capable of solving a more general problem in the services sector. This opens the possibility to develop rule systems that, whist mostly static, can react to different circumstances in the short term, and adapt in the medium to long term to maintain peak organisational effectiveness and efficiency.

Finally, the framework developed for the case study is capable of extension and improvement to cover other service business processes. And the essential idea that the decision should be given to the decision maker that is expected to get the most positive outcome for the organisation (not just making the right decision) could be applied to many decision-making situations.

12.6 Summary

The design and operation of business rules is a suitable subject for optimisation, and this has been applied this to a case study with CRA rules and operational adjustment. This takes account of the cost of obtaining information, the cost of processing, the profit from a good decision, and the loss from a bad decision. Internally there are three options: accept, reject, refer, request more information, but outwardly the process is binary where the final decision is to accept or reject).

Rules should be trained on outcomes rather than decisions to avoid expert bias and other sources of error. In this way, systems that include rules and experts can get better results than rules or experts alone. This system requires two levels of sophistication; the first level directs to cases through the rules, the second level decides on whether the case can be determined, requires further processing, or should be referred to an expert

The most effective technologies for good results, simplicity and reliability are decision trees, rule learning and logistic regression. Logistic regression has the advantage of estimating probabilities directly and dealing with complex cost structures and is a good technology as far as ways can be devised to deal with its shortcomings. However, some situations would use a hybrid approach where decision trees (or rule learning) create the business rules, and logistic regression is used to model the performance of the expert. Using both, we can calculate the most effective decide/continue/refer decision.

The framework devised has been created and tested using data from the financial services sector, but applies to many situations where:

- Decisions are based on data
- Rules, experts, or both can make decisions
- There is a cost in obtaining and processing that data
- Asking for too much data (of taking too long) results in lost opportunity
- There is a profit from the right decision and a loss from the wrong one

Finally, there are some limitations to this research insofar as it has (necessarily) focussed on one class of problems and case study examples. There are more problems, with more complexity that can be addressed.

13 REFERENCES

Aburto, L., & Weber, R. (2007). Improved supply chain management based on hybrid demand forecasts. Applied Soft Computing, 7(1), 136-144.

Adage, Expect to lose 20% of your audience within the first 10 seconds of playback, https://adage.com/article/digitalnext/marketing-online-video-viewers-quit-30-seconds/146218,

Aghdasi, M.; Malihi, S.E., (2010) Rule based business process optimization, Industrial Engineering and Engineering Management (IEEM), 2010 IEEE International Conference on , vol., no., pp.305,309

Agrawal, R., Pattanaik, L. N., & Kumar, S. (2012). Scheduling of a flexible job-shop using a multi-objective genetic algorithm. Journal of Advances in Management Research, 9(2), 178-188. doi:http://dx.doi.org/10.1108/09727981211271922

Aguilar-Saven, R. (2004). Business process modelling: Review and framework. International Journal of production economics 90.2: 129-149.

Ali, M. M., Törn, A., & Viitanen, S. (1997). A numerical comparison of some modified controlled random search algorithms. Journal of Global Optimization, 11(4), 377-385.

Alter, S. (2004). A work system view of DSS in its fourth decade. Decision Support Systems, 38(3), 319-327.

Amgoud, L., & Prade, H. (2009). Using arguments for making and explaining decisions. Artificial Intelligence, 173(3-4), 413-436.

Andreescu, A. I., & Uta, A. (2008). Methodological approaches based on business rules. *Informatica Economica Journal*, *12*(3), 23-27.

Antonius, P., & Farid, T. (2014). Management of Business Rules Approach-A model based on Resource Based View.

Apt, K. (2003). Principles of constraint programming. Cambridge university press.Australian Bureau of Statistics (ABS), Year Book Australia, 2012

Bahnsen, A. C., Aouada, D., & Ottersten, B. (2014, December). Example-dependent cost-sensitive logistic regression for credit scoring. In *2014 13th International Conference on Machine Learning and Applications* (pp. 263-269). IEEE.

Bajec, M., & Krisper, M. (2005). A methodology and tool support for managing business rules in organisations. Information Systems, 30(6), 423-443.

Baker, K. R., & Trietsch, D. (2013). Principles of sequencing and scheduling. John Wiley & Sons.

Beale, E. M. L. (1985). Integer programming. In Computational Mathematical Programming (pp. 1-24). Springer, Berlin, Heidelberg.

Begunov, N., Moskalev, I., & Klebanov, B. (2008, April). City agent-based model. In Proceedings of the 2008 Spring simulation multiconference (pp. 1-4).

Bensic, M., Sarlija, N., & Zekic-Susac, M. (2005). Modelling small-business credit scoring by using logistic regression, neural networks and decision trees. Intelligent Systems in Accounting, Finance & Management: International Journal, 13(3), 133-150.

Bhat, U. N. (2015). An introduction to queueing theory: modeling and analysis in applications. Birkhäuser.

Bilgen, B., & Ozkarahan, I. (2004). Strategic tactical and operational production-distribution models: a review. International Journal of Technology Management, 28(2), 151-171.

Bramer, M. (2007). Avoiding overfitting of decision trees. Principles of data mining, 119-134.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Belmont, CA: Wadsworth International

Brunswik, E. (1985). The essential Brunswik: beginnings, explications, applications, New directions in research on decision making. In Research conference on subjective probability, utility and decision making, Helsinki, Finland.

Bubeck, S. (2011). Introduction to online optimization. Lecture Notes, 2.

Bundy, A., Silver, B., & Plummer, D. (1985). An analytical comparison of some rule-learning programs. Artificial Intelligence, 27(2), 137-181.

Burns, R. B. (2000). Introduction to research methods. Frenchs Forest.

Castellanos, M., Casati, F., Umeshwar, D. and Ming-Chien, S. (2004), 'A Comprehensive and Automated Approach to Intelligent Business Processes Execution Analysis', Distributed and Parallel Databases, Vol. 16, pp. 1-35.

Cao, M., Stewart, A., & Leonard, N. E. (2010). Convergence in human decision-making dynamics. Systems & Control Letters, 59(2), 87-97.

Chang, S., Kim, S. D. O., & Kondo, G. (2015). Predicting default risk of lending club loans. Machine Learning, 1-5.

Chaston, I. (2017). The Service Sector. In *Technological Entrepreneurship* (pp. 215-239). Palgrave Macmillan, Cham.

Chen, Y. L., Wu, C. C., & Tang, K. (2009). Building a cost-constrained decision tree with multiple condition attributes. Information Sciences, 179(7), 967-979.

Chikofsky, E. J., & Cross, J. H. (1990). Reverse engineering and design recovery: A taxonomy. IEEE software, 7(1), 13-17.

Chisholm, M. (2004). How to build a business rules engine: extending application functionality through metadata engineering. Morgan Kaufmann.

Cibrán, M. A., D'hondt, M., & Jonckers, V. (2003, June). Aspect-oriented programming for connecting business rules. In Proceedings of the 6th International Conference on Business Information Systems (Vol. 6, No. 7, p. 24).

Cohen, W. W. (1995). Fast effective rule induction. In Machine learning proceedings 1995 (pp. 115-123). Morgan Kaufmann.

Cooray, K. (2006). Generalization of the Weibull distribution: the odd Weibull family. Statistical Modelling, 6(3), 265-277.

Čubrilo, M., Malekovi, M., & Rabuzin, K. (2016). Some Thoughts on Business Rules. The Journal of American Business Review, Cambridge, 4(2), 42.

Cuzzocrea, A., Decker, H., & Muñoz-Escoí, F. D. (2014, June). Scalable Uncertainty-Tolerant Business Rules. In International Conference on Hybrid Artificial Intelligence Systems (pp. 179-190). Springer, Cham.

Dantzig, G. B. (1998). Linear programming and extensions (Vol. 48). Princeton university press.

Beale, E. M. L. (1985). Integer programming. In Computational Mathematical Programming (pp. 1-24). Springer, Berlin, Heidelberg.

Wright, M. H. (1996). Direct search methods: Once scorned, now respectable. Pitman Research Notes in Mathematics Series, 191-208.D'Ariano, A., Pranzo, M., & Hansen, I. A. (2007). Conflict resolution and train speed coordination for solving real-time timetable perturbations. IEEE Transactions on intelligent transportation systems, 8(2), 208-222.

Davenport, T.H. (1993), Process Innovation: Reengineering Work Through Information Technology, Harvard Business School Press, Boston.

Dawson, C. (2019). Introduction to Research Methods 5th Edition: A Practical Guide for Anyone Undertaking a Research Project. Robinson.

Deb, R. K., & Serfozo, R. F. (1973). Optimal control of batch service queues. Advances in Applied Probability, 5(2), 340-361.

Dictionaries, C. (2009). Collins English Dictionary. HarperCollins Publishers.

Dormer, A. (2018, July). Business Rule Optimisation: Problem Definition, Proof-of-Concept and Application Areas. In International Conference on Business Information Systems (pp. 51-62). Springer, Cham.

Dorn, W. S. (1963). Non-linear programming—A survey. Management Science, 9(2), 171-208.

Drucker, P. (2018). The daily Drucker. Routledge.

Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

Eibe, Frank, Mark A. Hall, and Ian H. Witten (2016). The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques", Morgan Kaufmann, Fourth Edition, 2016.

Eiben, A. E., & Smith, J. E. (2003). Introduction to evolutionary computing (Vol. 53, p. 18). Berlin: springer.

Emekter, R., Tu, Y., Jirasakuldech, B., & Lu, M. (2015). Evaluating credit risk and loan performance in online Peer-to-Peer (P2P) lending. Applied Economics, 47(1), 54-70.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European journal of operational research*, *153*(1), 3-27.

Evans, M., Hastings, N., & Peacock, B. (2000). Statistical distributions.

Fan, Y.S. (2001), Fundamental of Workflow Management Technology, Springer-Verlag, New York.

Federal Deposit Insurance Corporation (FDCI), (2015). Risk Management Examination Manual for Credit Card Activities, Chapter VII, FDIC- Division of Supervision and Consumer Protection

Freitas, A., Costa-Pereira, A., & Brazdil, P. (2007, September). Cost-sensitive decision trees applied to medical data. In International Conference on Data Warehousing and Knowledge Discovery (pp. 303-312). Springer, Berlin, Heidelberg.

Fürnkranz, J., Gamberger, D., & Lavrač, N. (2012). *Foundations of rule learning*. Springer Science & Business Media.

Galindo, J., & Tamayo, P. (2000). Credit risk assessment using statistical and machine learning: basic methodology and risk modeling applications. Computational Economics, 15(1-2), 107-143.

Gartner, (2012). Vendors in the Business Rule Market, ID Number: G00226647

Gibillini, M. (2008). Beyond process optimization. Canadian Underwriter, 75(1), 46-46,48.

Gill, P. E., Murray, W., & Saunders, M. A. (2005). SNOPT: An SQP algorithm for large-scale constrained optimization. SIAM review, 47(1), 99-131.

Gottesdiener, E. (1997). Business rules show power, promise. Application Development Trends, 4(3), 36-42.

Guéret, C., Prins, C., & Sevaux, M. (2000). Applications of optimization with Xpress-MP.

Gummesson, E. (2000). Qualitative methods in management research. Sage.

Gunnell, D., Middleton, N., Whitley, E., Dorling, D., & Frankel, S. (2003). Why are suicide rates rising in young men but falling in the elderly?—a time-series analysis of trends in England and Wales 1950–1998. Social science & medicine, 57(4), 595-611.

Gupta, A. K., Lotlikar, R. M., & Rai, A. (2013). U.S. Patent Application No. 13/177,998.

Gupta, A. K., Lotlikar, R. M., & Rai, A. (2013). U.S. Patent Application No. 13/594,963.

Hall, M. A., & Holmes, G. (2003). Benchmarking attribute selection techniques for discrete class data mining. IEEE Transactions on Knowledge and Data engineering, 15(6), 1437-1447.

Hall, M. A. (2000). Correlation-based feature selection of discrete and numeric class machine learning.

Hall, M. (1999). Correlation-based Feature Subset Selection for Machine Learning. PhD thesis, University of Waikato.

Hall, Z., Batta, R., & Szczerba, R. (2001). Supply-Chain Optimisation: -players, tools and issues. OR Insight, 14(2), 20-30.

Harmon, P. (2008, September). Business process management: Today and tomorrow. In International Conference on Business Process Management (pp. 1-1). Springer, Berlin, Heidelberg.

Harte, J. M., & Koele, P. (2001). Modelling and describing human judgement processes: The multiattribute evaluation case. Thinking & reasoning, 7(1), 29-49.

Hay, D., & Anderson Healy, K. (2000). Defining Business Rules ~ What Are They Really? The Business Rules Group, Final Report, revision 1.3

Hegazi, M. O. (2015). Measuring and Predicting the Impacts of Business Rules Using Fuzzy Logic. International Journal of Computer Science and Information Security, 13(12), 59.

Hosmer Jr, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied logistic regression (Vol. 398). John Wiley & Sons.

Hunt, V. D. (1996). Process mapping: how to reengineer your business processes. John Wiley & Sons.

Jandir, R. (2009). Event based propagation approach to constraint configuration problems.

Janssen, F., & Fürnkranz, J. (2010). On the quest for optimal rule learning heuristics. Machine Learning, 78(3), 343-379.

Johanson, H.J., McHugh, P., Pendlebury, A.J. and Wheeler, W.A.I. (1993), Business Process Reengineering - Breakpoint Strategies for Market Dominance, Wiley, Chichester.

Kaggle (2020) https://www.kaggle.com/wendykan/lending-club-loan-data

Kamada, A., & Mendes, M. (2007, October). Business rules in a service development and execution environment. In 2007 International Symposium on Communications and Information Technologies (pp. 1366-1371). IEEE.

Kamrani, Farzad; Ayani, Rassul; Moradi, Farshad , A framework for SIMULATION-based optimization of business process models, Simulation, ISSN 0037-5497, 07/2012, Volume 88, Issue 7, pp. 852 – 869

Kardasis, P., & Loucopoulos, P. (2004). Expressing and organising business rules. Information and Software technology, 46(11), 701-718.

Karelaia, N., & Hogarth, R. M. (2008). Determinants of linear judgment: A meta-analysis of lens model studies. Psychological bulletin, 134(3), 404.

Kaufmann, E., Reips, U. D., & Wittmann, W. W. (2013). A critical meta-analysis of lens model studies in human judgment and decision-making. PloS one, 8(12).

Kimmig, A., Demoen, B., De Raedt, L., Costa, V., & Rocha, R. (2011). On the implementation of the probabilistic logic programming language ProbLog. Theory and Practice of Logic Programming, 11(2-3), 235-262. doi:10.1017/S1471068410000566

Kirk, D. E. (2004). Optimal control theory: an introduction. Courier Corporation.

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. Artificial intelligence, 97(1-2), 273-324.Koller, D., & Sahami, M. (1996). Toward optimal feature selection. Stanford InfoLab.

Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. Emerging artificial intelligence applications in computer engineering, 160, 3-24.

Kotsiantis, S. B. (2013). Decision trees: a recent overview. Artificial Intelligence Review, 39(4), 261-283.

Kruk, J., Quigney, P., Tehrani, S., Richards, P., & Shaver, J. (2003). U.S. Patent Application No. 10/279,138.

Kunz, T. P., & Crone, S. F. (2015). The impact of practitioner business rules on the optimality of a static retail revenue management system. Journal of Revenue and Pricing Management, 14(3), 198-210.

Jaillet, P., & Wagner, M. R. (2010). Online Optimization—An Introduction. In Risk and Optimization in an Uncertain World (pp. 142-152). INFORMS.

Laguna, M., & Marklund, J. (2013). Business process modeling, simulation and design. CRC Press.

Lahsasna, A., Ainon, R. N., & Teh, Y. W. (2010). Credit Scoring Models Using Soft Computing Methods: A Survey. Int. Arab J. Inf. Technol., 7(2), 115-123.

Lawler, E. L., Lenstra, J. K., Kan, A. H. R., & Shmoys, D. B. (1993). Sequencing and scheduling: Algorithms and complexity. Handbooks in operations research and management science, 4, 445-522.

Lend Academy, (2011) How Lending Club and Prosper Set Interest Rates, 2011, https://www.lendacademy.com/how-lending-club-and-prosper-set-interest-rates/

Leonardi, M. C., & Leite, J. C. S. (1998, April). Business Rules as Organizational Policies. In Software Specification and Design, International Workshop on (pp. 68-68).

Liao, S. H. (2005). Expert system methodologies and applications—a decade review from 1995 to 2004. Expert systems with applications, 28(1), 93-103.

Lin, G., Shen, C., Shi, Q., Van den Hengel, A., & Suter, D. (2014). Fast supervised hashing with decision trees for high-dimensional data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 1963-1970).

Liu, F., Hilgers, N., Nelsen, M., Siegel, K., Brown, C., & Dwyer, T. (2014). U.S. Patent No. 8,924,279. Washington, DC: U.S. Patent and Trademark Office.

Lomax, S., & Vadera, S. (2013). A survey of cost-sensitive decision tree induction algorithms. ACM Computing Surveys (CSUR), 45(2), 1-35.

Maciejewski, M. (2014). Online taxi dispatching via exact offline optimization. Logistyka, 3, 2133-2142.

Mägi, I., (2016). Performance Causing Users to Abandon Site, https://plumbr.eu/blog/user-experience/performance-causing-users-abandon-site

Mark, H., Ian, W., & Eibe, F. (2011). Data mining: practical machine learning tools and techniques. In Morgan Kaufmann Publishers.

Matousek, J., & Gärtner, B. (2007). Understanding and using linear programming. Springer Science & Business Media.

Mousavi, A., & Sellers, E. (2019). Optimisation of production planning for an innovative hybrid underground mining method. Resources Policy, 62, 184-192.

Ninan, J., Van Gorp, P. M. E., Reijers, H. A., & Vdovjak, R. (2014). Integrating rules and automated planning in business processes.

Object Management Group, (2011). Business Process Model and Notation (BPMN). Version 2.0,

Osman, M. H. B., Kaewunruen, S., & Jack, A. (2018). Optimisation of schedules for the inspection of railway tracks. Proceedings of the Institution of Mechanical Engineers, Part F: Journal of Rail and Rapid Transit, 232(6), 1577-1587.

Oswald, A. (2013). Measuring the impact of business rules on inventory balancing (Doctoral dissertation, Monterey, California: Naval Postgraduate School).

Packt, (2009). Business Rules Management, BPM, and SOA, www.packtpub.com/article/business-rules-management-bpm-and-soa

Pinto, R., Mettler, T., & Taisch, M. (2013). Managing supplier delivery reliability risk under limited information: Foundations for a human-in-the-loop DSS. Decision support systems, 54(2), 1076-1084.

Ponstein, J. P. (2004). Approaches to the Theory of Optimization (Vol. 77). Cambridge University Press.

Quinlan, R., (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.

Quinzaños, J. M., Cartas, A., Vidales, P., & Maldonado, A. (2014, May). iDispatcher: Using Business Rules to Allocate and Balance Workloads. In DSS (pp. 110-119).

Rajamma, R. K., Paswan, A. K., & Hossain, M. M. (2009). Why do shoppers abandon shopping cart? Perceived waiting time, risk, and transaction inconvenience. Journal of Product & Brand Management.

Ratkowski, A., Sacha, K., & Zalewski, A. (2012, September). Optimization of business processes in service oriented architecture. In 2012 IEEE 16th International Enterprise Distributed Object Computing Conference Workshops (pp. 42-50). IEEE.

Robson, C. (2002). Real World Research Oxford: Blackwell.

Roger, A. E., Marcel, F. N., & Lopez, A. C. (2010). A model based business process requirement rule specification. International Journal of Computer Applications, 11(9)

Rosca, D., & Wild, C. (2002). Towards a flexible deployment of business rules. Expert Systems with Applications, 23(4), 385-394.

Ross, R. G. (2006). The RuleSpeak business rule notation. Business Rules Journal, 7(4).

Ross, R. (2016). Business Rules–What You Need to Know. Business Rules Journal.

Saidi-Mehrabad, M., & Fattahi, P. (2007). Flexible job shop scheduling with tabu search algorithms. The international journal of Advanced Manufacturing technology, 32(5-6), 563-570.

Schauer, J., & Schwarz, C. (2013). Job-shop scheduling in a body shop. Journal of Scheduling, 16(2), 215-229.

Schernhammer, E. (2005). Taking their own lives—the high rate of physician suicide. N Engl J Med, 352(24), 2473-6.

Schreiber, A. T., Schreiber, G., Akkermans, H., Anjewierden, A., Shadbolt, N., de Hoog, R., ... & Wielinga, B. (2000). Knowledge engineering and management: the CommonKADS methodology. MIT press.

Schrijver, A. (1998). Theory of linear and integer programming. John Wiley & Sons.

Schulte, R. Clark, W. Driver, M. Dunie, R, (2018), Innovation Tech Insight for Business Rules Management Systems, Gartner, G00354638

Serrano-Cinca, C., Gutiérrez-Nieto, B., & López-Palacios, L. (2015). Determinants of default in P2P lending. PloS one, 10(10).

Shao, J., & Pound, C. J. (1999). Extracting business rules from information systems. BT Technology Journal, 17(4), 179-186.

Shen, H., Wall, B., Zaremba, M., Chen, Y., & Browne, J. (2004). Integration of business modelling methods for enterprise information system analysis and user requirements gathering. Computers in Industry, *54*(3), 307-323.

Sneed, H. M., & Erdos, K. (1996, March). Extracting business rules from source code. In WPC'96. 4th Workshop on Program Comprehension (pp. 240-247). IEEE.

Sox, C. R., Jackson, P. L., Bowman, A., & Muckstadt, J. A. (1999). A review of the stochastic lot scheduling problem. International Journal of Production Economics, 62(3), 181-200.

Sprague Jr, R. H. (1980). A framework for the development of decision support systems. MIS quarterly, 1-26.

Steen, B., Pires, L. F., & Iacob, M. E. (2010, October). Automatic generation of optimal business processes from business rules. In 2010 14th IEEE International Enterprise Distributed Object Computing Conference Workshops (pp. 117-126). IEEE.

Stelling, M., Roy, R., Tiwari, A., & Majeed, B. (2006). A coding mechanism for business process optimisation using genetic algorithmS: Initial thoughts. In Artificial Intelligence and Soft Computing (pp. 312-316).

Stelling, M., Roy, R., & Tiwari, A. (2009). A novel modelling and optimisation technique for business processes: an evolutionary computing-based approach. In Applications of Soft Computing (pp. 75-85). Springer, Berlin, Heidelberg.

Subramania, H. S., & Khare, V. R. (2011). Pattern classification driven enhancements for human-in-the-loop decision support systems. Decision Support Systems, 50(2), 460-468.

Talbi, E. G., Rahoual, M., Mabed, M. H., & Dhaenens, C. (2001, March). A hybrid evolutionary approach for multicriteria optimization problems: Application to the flow shop. In International Conference on Evolutionary Multi-criterion Optimization (pp. 416-428). Springer, Berlin, Heidelberg.

Tarasofsky, J. (2008). Online Reservations: Increasing Your Site's 'Look to Book'Ratio. Hotel Marketer.

Taylor, J. (2011). Decision management systems: a practical guide to using business rules and predictive analytics. Pearson Education.

Technopedia. (2011). Abandonment. www.techopedia.com/definition/1387/abandonment

TechTarget (2017). Knowledge Engineering www.searchenterpriseai.techtarget.com/definition/knowledge-engineering.

Timofeev, R. (2004). Classification and regression trees (CART) theory and applications. Humboldt University, Berlin, 1-40.

Tiwari, A., Hoyos, P. N., Hutabarat, W., Turner, C., Ince, N., Gan, X. P., & Prajapat, N. (2015). Survey on the use of computational optimisation in UK engineering companies. CIRP Journal of Manufacturing Science and Technology, 9, 57-68.

Tulasi, C. L., & Rao, A. R. (2012). Review on theory of constraints. International Journal of Advances in Engineering & Technology, 3(1), 334.

Turney, P. D. (1994). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. Journal of artificial intelligence research, 2, 369-409.

UK Cabinet Office, (2012). Government Digital Strategy

US Labour Statistics (2010). http://www.bls.gov/ooh/office-and-administrative-support/customer-service-representatives.htm http://www.bls.gov/ooh/business-and-financial/loan-officers.htm http://www.bls.gov/ooh/business-and-financial/insurance-underwriters.htm
http://www.bls.gov/ooh/management/financial-managers.htm

van der Geest, T. T., & Pruyn, A. A. (2016). Online Shopping Abandonment Rate.

Van Eijndhoven, T., Iacob, M. E., & Ponisio, M. L. (2008, September). Achieving business process flexibility with business rules. In 2008 12th International IEEE Enterprise Distributed Object Computing Conference (pp. 95-104). IEEE.

Vergidis, K., Saxena, D., & Tiwari, A. (2012). An evolutionary multi-objective framework for business process optimisation. *Applied Soft Computing*, *12*(8), 2638-2653.

Vergidis, K., Tiwari, A., & Majeed, B. (2007). Business process analysis and optimization: Beyond reengineering. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 38(1), 69-82.

Vergidis, K. (2008). Business process optimisation using an evolutionary multi-objective framework.

Ververidis, D., & Kotropoulos, C. (2005, September). Sequential forward feature selection with low computational cost. In 2005 13th European Signal Processing Conference (pp. 1-4). IEEE.

Visentini, M. S., Borenstein, D., Li, J. Q., & Mirchandani, P. B. (2014). Review of real-time vehicle schedule recovery methods in transportation services. *Journal of Scheduling*, *17*(6), 541-567.

Vysockis, T., & Vasilecas, O. (2018). Simulation Based Multi-Objective Optimization for Dynamic Business Processes. In Doctoral Consortium/Forum@ DB&IS (pp. 142-150).

Wallace, M. (2020). *Building Decision Support Systems: using MiniZinc*. Springer Nature.

Wallenius, P. K. J. (1992). Multiple criteria decision support.

Wang, W., Indulska, M., & Sadiq, S. (2014). Integrated modelling of business process models and business rules: a research agenda. ACIS.

Wang, O., Liberti, L., D'Ambrosio, C., de Sainte Marie, C., & Ke, C. (2016, July). Controlling the average behavior of business rules programs. In International Symposium on Rules and Rule Markup Languages for the Semantic Web (pp. 83-96). Springer International Publishing.

Wang, W. (2017). Integrated Modeling of Business Processes and Business Rules.

Wei, W., Indulska, M., & Sadiq, S. (2017). Guidelines for Business Rule Modeling Decisions. Journal of Computer Information Systems, 1-11.

Weiner, M. (2004). Six Sigma applied research for improved public relations. Communication World-San Francisco-, 21(1), 26-29.

Witten, I. H., Frank, E., Hall, M. A., & Pal, C. J. (2016). Data Mining: Practical machine learning tools and techniques. Morgan Kaufmann.

Wolfe, P. (1959). The simplex method for quadratic programming. Econometrica: Journal of the Econometric Society, 382-398.

Wright, M. H. (1996). Direct search methods: Once scorned, now respectable. Pitman Research Notes in Mathematics Series, 191-208.

Xin, D., Ma, L., Liu, J., Macke, S., Song, S., & Parameswaran, A. (2018, June). Accelerating human-in-the-loop machine learning: Challenges and opportunities. In Proceedings of the Second Workshop on Data Management for End-To-End Machine Learning (p. 9). ACM.

Xu, Y., & Huang, J. S. (2015). Factors influencing cart abandonment in the online shopping process. Social Behavior and Personality: an international journal, 43(10), 1617-1627.

Yang, X. (2008). Introduction to mathematical optimization. From Linear Programming to Metaheuristics.

Yan, J., Gao, H., & Mu, Y. (2015, June). Business Rule Driven Composite Service Optimisation and Selection. In Services Computing (SCC), 2015 IEEE International Conference on(pp. 49-56). IEEE.

Zadrozny, B., Langford, J., & Abe, N. (2003, November). Cost-sensitive learning by cost-proportionate example weighting. In Third IEEE international conference on data mining (pp. 435-442). IEEE.

Zimmerman, K., & Javits, N. (2016). U.S. Patent Application No. 15/164,272.

14 APPENDICES

14.1 Context

In 5.2.4 we discussed the problem of hiring a person to do a job. We could address this problem in two ways: offer a salary and then choose an applicant or choose an applicant and the adjust the salary to attract the desired applicants (or applicants). The problem changes from a choice decision (binary or integer – who to accept) to real decision (how much to offer). This is essentially a dual optimisation problem.

14.2 Primal and Dual Problems

The classical expression of the dual problem is that of maximising the contribution of a set of resources given a constraint on the overall cost of engagement (Matousek et al, 2007).

Suppose we have a set of resources (people) x_i $1 \leq i \leq n$ and we can express their economic contribution as $c^T x$ and the cost of their employment as Ax where c is an n-vector and A is a n x m matrix. We have limits on the cost of employment defined by b_j $1 \leq j \leq m$. Costs could include salaries of each employee and overheads.

The resource allocation problem is defined as: maximise the economic contribution with an overall limit on the cost. This can be expressed as:

$$\max_x c^T x \colon Ax \leq b, x \geq o$$

We can then form the dual problem based on the dual variable y_j $1 \leq j \leq m$ and we have:

$$\min_y b^T y \colon A^T y \geq c$$

This is the resource valuation problem and the interpretation is: minimise the cost whilst spending enough to achieve the economic outcome.