Computational intelligence techniques for general insurance

Pietro Parodi

November 19, 2009

Contents

Abstract

This paper is an attempt to answer the question "What is the proper framework for understanding risk?" in the context of general insurance. It argues that although actuaries and other risk professionals tend to deal with risk in the context of classical statistics and by resorting to subjective judgment to compensate the inadequacies of this framework, understanding risk is actually an "ecological" problem and it is more fruitful to look at risk in the context of computational intelligence.

Computational intelligence (a.k.a. artificial intelligence) is the discipline that deals with designing intelligent agents – agents that are able to learn from data, integrate multiple sources of (often uncertain) knowledge, deal with changes in an environment, make decisions, compete against each other. These are much the same problems that players in the insurance market, whether they be customers, insurers, reinsurers, or the regulator, face daily – and therefore it is quite natural to describe these players as *risk agents* that have to survive and thrive in an environment (the market).

Most actuarial problems, such as pricing, reserving, capital modelling, DFA, pricing optimisation, can be naturally framed as computational intelligence problems. This is not only a theoretical issue: it also allows a technological transfer from the computational intelligence discipline to general insurance, wherever techniques have been developed for problems which are common to both contexts. This has already happened, at least in part, as some computational intelligence techniques such as neural networks, k-means clustering, data mining have found a place in actuarial literature and practice. Others (such as sparsity-based regularisation or dynamic decision networks) are perhaps less familiar to risk professionals, and one of the goals of this paper is indeed to "fill the gaps", introducing cutting-edge techniques such as the elastic net or spectral clustering to the wider actuarial community.

In this paper, computational intelligence techniques are always illustrated through simple insurance applications and are systematically compared to one another to determine their adequacy to our needs: regularisation, neural networks and generalised linear modelling are compared as tools for predictive modelling; fuzzy set theory and Bayesian analysis are compared as tools for dealing with uncertain and soft knowledge; dynamic Bayesian networks

and Kalman filtering are compared as tools to understand changes in the environment; and so on.

Apart from the methodological findings, some practical recommendations also emerge from this investigation: (i) that the use of regularisation techniques should be expanded in actuarial practice, as these solve the problem of variable selection efficiently; (ii) that cross-validation and the Bayesian Information Criterion should be used for model validation alongside more traditional methods such as the Akaike Information Criterion and the hold-out sample method, especially in situations where data is scant; (iii) that Bayesian analysis should be the preferred tool for dealing with data uncertainty, model uncertainty and soft knowledge (rather than fuzzy set theory or rule-based systems), and that Bayesian networks should be used where we have complex chains of dependency; (iv) that dynamic Bayesian networks are a more general alternative to Kalman filtering (applied in the past to reserving and pricing) to deal with environment changes, and can also be extended to incorporate decisions and utility (dynamic decision networks), allowing to deal with problems such as DFA and pricing optimisation; (v) that multi-agent systems may be used to simulate markets and design optimal strategies in the face of competition, or to design regulation that achieves some overall goal. In all these techniques, the Bayesian framework is ubiquitous.

Computational intelligence techniques are powerful and (unlike artificial intelligence in the 70s or 80s) are now on a more rigorous basis. However, there are some obvious limitations, which boil down to the failing of artificial intelligence to achieve its goals stated in the early manifestos in the 1960s and beyond: that of producing agents which exhibited truly intelligent behaviour. None of the techniques above truly replace human judgment, but rather support it and enhance our ability to justify decisions quantitatively. A, when they are used for prediction they work only when the environment exhibits some sort of stationarity – and the non-stationarity introduced by humans changing the rules of the game is especially difficult to tackle. Finally, some of the techniques – especially those involving multiple agents – are so complex and rich in variables that they often offer only a formal, rather than practical, solution.

Acknowledgments

I would like to thank the profession for giving me the opportunity to carry out a piece of research that was in my mind since I have started working as an actuary, but never had the time to pursue. In particular I would like to thank Julian Lowe, my internal supervisor, for his time and guidance and especially for the kind encouragement throughout the project.

I would like to thank Aon Benfield for providing the data that were used in some of the applications, and for giving me leave to carry out this project.

I am indebted to Alessandro Verri for giving me the opportunity of spending two weeks at the University of Genoa. Thanks to his hospitality and the mind-opening conversations we had, these two weeks were an enjoyable return to academia after several years. My thanks extend to his team, for talking me through their projects and for their precious feedback on this work. I am especially thankful to Nicola Rebagliati, who ran the experiments on IBNER clustering and turned me into an instant fan of spectral clustering; to Annalisa Barla, for the time she spent helping me with the regularisation software, and to Curzio Basso for organising my talks at the University of Genoa.

I would like to thank these people for very helpful discussion: Ernesto De Vito (from whom I borrowed his elegant treatment of regularisation), Vito Fragnelli (who introduced me to unexpected applications of game theory to insurance), Viviana Mascardi (who got me started on multi-agent systems), Jill Boulton (who had an expert word on virtually all topics we touched upon), Stephen Roberts (for some state-of-the-art conversation on partially observable Markov decision processes), Ji Yao (for shedding light on his work on territories clustering).

I would like to thank my colleagues at Aon Benfield for their feedback and advice during my presentations to them, and their keen interest in this project.

Finally, a big thank you to my wife Francesca and my family for putting

up with so many marred weekends and many missed visits to Colchester zoo. Hopefully there will be less time spent with risk agents and more with elephants in the near future!

Chapter 1

Introduction

> First law: You can't win. Second law: You can't break even.
> Third law: You can't quit the game.
> *The three laws of thermodynamics, according to C.P.Snow.*

Risk professionals such as underwriters, traders, risk managers, actuaries, quants, and many others, have to understand risk in order to do their job effectively. To carry out their work they have a variety of tools at their disposal: annuity tables, derivatives, exposure curves, catastrophe models, extreme value theory, and so on. In some areas, such as life insurance, these tools are quite established and reliable: although research on life insurance is still being carried out, its foundations are solid. In investment, stochastic differential equations provide a framework in which to analyse option pricing. In other areas, such as general insurance, there is much less consensus on what techniques should be used for what, and the results obtained by different techniques may vary widely. Furthermore, there is no overall framework to address general insurance problems.

This paper represents the author's attempt to answer the question: ”What is the appropriate framework to describe and understand risk?”, at least in the context of general insurance. The investigation has stemmed from the observation that although many actuarial problems are framed as classical statistics problems, in practice the actuary is taught never to solve problems in that way, especially where data are not overwhelming. Rather, the actuary is taught to exercise judgment constantly.

Consider the classic problem of calculating the prospective loss ratio for a line of business based on the underwriting results over a number of years.

This seems simple enough – calculate the weighted average of claims divided by premiums over a number of years, and use that as a prospective loss ratio – but we know that a lot of caveats must be applied before we do that. Premiums need to be brought to current terms by applying past rate changes. Past inflation needs to be applied to claims from previous years in order to bring them to their current value. Some amount of IBNR may have to added to the most recent years, especially if the account has some long-tail component (e.g. large liability losses). Underwriting changes and business mix changes need to be taken into account – if an insurance company was writing a very risky account 10 years ago (e.g. a large percentage of young drivers for a motor account) and has now become stricter and only has a very limited amount of risky drivers, it may well have to make some corrections to past experience to "as-if" it to today's terms. Once the actuary or the underwriter has done all that, an average can actually be computed which is more relevant to today's needs. At that point the thorniest issue of all arises, namely is the past experience actually relevant to tomorrow's environment? At this point the actuary can do no better than to include all information available on how the world is changing and consider possible future scenarios. What seemed quite a harmless statistical exercise for which a high-school kid could easily work out a quick solution has rapidly become a different problem altogether, where the underwriter basically needs to be able to:

- estimate relevant past inflation (what to use?);
- estimate the impact of underwriting changes/business mix changes/legislation changes/policy terms/... to past claims;
- estimate the amount of IBNR, IBNER, etc...;
- consider how the environment has changed and might reasonably change in the near future.

All of a sudden this is a problem (of which we have given here quite a domesticated version) for grownups, and streetwise grownups at that.

Actually, is this still a statistical problem at all? It seems that if we want a reasonable answer we have somehow to take into account a lot of information that is mired in uncertainty – the relevant past inflation is almost never something we can be sure about, IBNR is of course only an estimate for which dozens of different methods give different results, and the effect of underwriting changes, effect of policy mix, impact of legislation are also only a rough estimate, apart from some very fortunate cases.

Although it is difficult to get the prospective loss ratio right, it is very easy to get it wrong if we do not take into account all the information we have. In a case like this, there is plenty of information which can neither be rigorously treated nor ignored, especially if one wants to avoid losing money.

1.1 Some risk epistemology

The issue is that understanding risk requires having an effective model of the environment you are moving in. Not necessarily a mathematical model – perhaps not even an explicit model, but a model nonetheless. If you are pricing property insurance, you must have a model of what could cause damage to properties – ultimately, you need to understand to a certain degree how floods and earthquakes happen. You also need to understand that in a recession some policyholders will exhibit a stronger inclination to arson. You need to understand what the market for property insurance is, and what your competitors are doing in terms of pricing and everything else. Knowing all this will not take the randomness of the business away, but will make you far better equipped to be successful.

The problem of understanding risk is therefore an ecological problem rather than a mathematical or scientific one, in the sense that ultimately, you want to *survive* and actually to *thrive* in a world that competes very harshly for resources and where the rules of the game changes constantly. You want to get the price of your policies right so that you can have a decent market share and make a profit, in a world where everybody else is doing the same and resources (including time) are limited.

On the up side (at least for actuaries), the environment is not the jungle but a mathematically sophisticated one. Producing good statistics certainly helps to have a competitive edge. It is just not good enough.

1.2 Understanding risk in general insurance

So far I have shied away from defining what risk is. Although everybody agrees that risk has something to do with uncertainty regarding the outcome of future (adverse) events, trying to come up with a precise definition of risk is an unnecessary distraction[1]. As physics has been defined by Popper "simply what physicists do", we are in very good company if we assume that risk is what risk professionals work on. For the purpose of this paper, risk will actually be what general insurance actuaries, underwriters, claims managers work on.

In this admittedly limited general insurance context, understanding risk involves concretely:

- making predictions based on data ("learning from data"), e.g. estimating the frequency and severity parameters of a loss distribution, selecting rating factors, estimating reserves, estimating the 99.5% value at risk for capital requirements...;
- dealing with limited, uncertain and soft/expert knowledge, e.g. individual loss estimates;
- dealing with risk that changes with time, and with the integration of new knowledge, e.g. incorporating new information when doing a reserve exercise;
- making successful decisions in an uncertain environment, e.g. setting the right price for a policy, running a DFA exercise, produce a business plan;
- modelling collective behaviour: markets, competition, etc, e.g. again setting the right price for a policy given (limited) knowledge of what the competitors do, or designing adequate regulation on capital requirements.

All the above are typical problems of computational intelligence. Computational intelligence is just another word for artificial intelligence – perhaps with less of a science fiction flavour, and for this reason a more popular choice among the sceptics. It is that discipline that attempts to design *intelligent agents*. Of all definitions one wants to avoid, that of "intelligent" is probably at the top of the list (see, however, [RN03] for a discussion): suffice it to say that an intelligent agent is anything that is able to perceive its environment through sensors and to act upon that environment. An example is that of

[1] Karl Popper has argued that the quest for definitions (or asking questions such as "What do you mean by...?") is *always* a distraction, and a trademark obsession of linguistic philosophy and of essentialism.

artificial fish in an artificial sea – researchers have created wonderful examples of artificial life where each fish is equipped with “a perception system to detect their surroundings, a motor system to control movements, and a behavior system to coordinate relevant actions” (see for example [TTG94]).

Another example, undoubtedly more relevant to us, is that of an insurance company which has to survive and thrive in the insurance market. It receives information about the losses it makes and about how its competitors are pricing similar business and based on that it devises a strategy, makes decisions on prices, reserves, capital to be held (within regulatory constraints).

The analogy between fish and insurance companies can only be pursued up to a point: for example insurance companies do not perform mating dances. However, it turns out that many of the techniques devised for artificial fish and intelligent agents in general apply well to the insurance company above or in general to anybody making rational decisions in an uncertain environment. The list describing what "understanding risk" involves concretely in general insurance is not a carefully chosen selection of topics which happen to be common to both general insurance and computational intelligence: they basically cover the whole discipline of computational intelligence. For example, here is the list of contents of the central part of the book by Russell and Norvig "Artificial Intelligence: A Modern Approach" [RN03], arguably the best reference for the discipline:

- `11. Planning`
- `12. Planning and acting in the real world`
- `13. Uncertainty`
- `14. Probabilistic reasoning`
- `15. Probabilistic reasoning over time`
- `16. Making simple decisions`
- `17. Making complex decisions`
- `18. Learning from observations`
- `19. Knowledge in learning`
- `20. Statistical learning methods`

Apart from some of the titles which may sound a bit abstract, I think risk professionals will find that much of their work is reflected in this list.

1.3 Alternative frameworks

In replying to the question "What is the appropriate framework for understanding risk?" one must of course be aware of what the current framework is and what alternatives there exist to it[2]. We have already started replying to this question at the beginning of this introduction, stating that risk professionals have a number of techniques at their disposal to understand risk, but no common framework.

To elaborate a little bit more on the question of the current framework, it is of course a difficult question to answer because different risk practitioners will have different views on the subject. A cue may be found in the way actuaries are trained to become professionals. Traditionally, and across several regulatory bodies (certainly in the UK, the US and in Australia), there is a two-stage exam system: in the first stage a number of technical exams ensuring that the actuary has a solid understanding of the basic actuarial techniques. Most of these techniques are based on calculus, probability theory and statistics. In the successive stage, the actuary learns more specialised techniques which are relevant to his/her specialisation, but is also encouraged to exercise judgment which goes beyond what can be proven quantitatively, and which is partly acquired through study material on market practice, and partly through skills learned in the workplace. It is fair to say, therefore, that the assumed framework for understanding risk is classical probability/statistics (with a sprinkling of Bayesian statistics) enhanced by the ability of making good judgment calls based on market knowledge and professional experience.

In a nutshell, an actuary will first be taught how to use his actuarial tools, and subsequently he will be taught not to trust them too much – and not to blame them for his poor understanding of risk.

1.4 The purpose of this study

The purpose of this investigation is twofold:

1. One is methodological, bordering on the philosophical: argue that the "risk agent" paradigm is (currently) the most promising framework for

[2] I am indebted to Warren Dresner for pointing out the importance of addressing this question.

describing and understanding risk, at least in the context of general insurance. This will be achieved by identifying typical problems of general insurance and showing how they can be described and solved by computational intelligence techniques. Simple practical examples from the literature will be used where possible – the examples will hopefully be both very simple and reasonably realistic.

2. The other is more practical: show what we as risk professionals can learn from the computational intelligence community when performing our daily taks such as pricing, reserving, capital modelling. This will hopefully be achieved by providing a review of computational intelligence techniques, including techniques which might be new to a part of the actuarial community, or by comparing industry standards (e.g. GLM) with their competitors in machine learning (e.g. neural networks).

1.5 From prior to updated beliefs

This investigation has started with a number of preconceptions in mind:

- that no unifying framework could be found, but that many computational intelligence would have turned out to be useful for specific tasks. This is reflected in the title of this dissertation, which focuses on "techniques" rather than on a "framework".
- that computational intelligence was a collection of clever heuristics, without a centre of gravity;
- that fuzzy set theory was the most promising tool to deal with uncertain and soft knowledge (especially data uncertainty), or at least a valid alternative to probabilistic reasoning;
- that neural networks might give interesting results but they lack transparency;
- that Kalman filtering is the most appropriate tool to analyse changes that vary in time;
- that multi-agent systems would provide a possible framework to analyse collective behaviour – along with graph theory and artificial life experiments.

Some of these preconceptions (my view on black-box methodologies, for example) I have retained and inform my treatment of technologies such as neural networks. Other have survived in a mutated form. However, some of these preconceptions were drastically revised as a result of this investigation:

- Computational intelligence has made significant progress in the last 10-15 years, and is now far more than a collection of unrelated heuristics. A remarkable systematisation work on machine learning has been carried out by the Stanford school of statisticians (Tibshirani, Hastie, Efron, Zou, Friedman . . .). The book by three of them (Hastie, Tibshirani and Friedman), "The Elements of Statistical Learning" (2001), has become a classic. Another more general book, that by Russell and Norvig (2003), has changed the way we now look at Artificial Intelligence as a discipline.

- As far as a unifying framework is concerned, there might not be one on which everybody agrees, but the Bayesian approach has become pervasive, thanks especially to the increase of computational power and introduction of efficient numerical techniques (MCMC, Gibbs sampling . . .). This is especially true for those aspects of computational intelligence that are of closest interest to risk professionals – that is, the ones that allow to quantify risk, rather than the linguistic/logical aspects of computational intelligence that are still the focus of many practitioners in that field.

- Specifically, Bayesian analysis turns out to be a far more powerful tool than fuzzy set theory to describe and analyse uncertainty.

- Once one embraces the Bayesian framework, one finds a goldmine of methods to address many of the problems discussed above: Bayesian networks to analyse complex chains of dependencies and uncertainties; dynamic Bayesian networks (DBNs) to analyse information that varies with time – a more general tool than Kalman filtering; and dynamic decision networks, which are a good framework for systems that have to make decisions and therefore provide a good representation of intelligent agents and the building blocks for multi-agent systems to analyse the collective behaviour of markets.

1.6 How this paper is organised

In writing a paper like this, which aims at spotting cross-fertilisation opportunities between two different disciplines, there is always the risk of doing a poor job of integrating techniques (from computational intelligence) and applications (from general insurance). I have tried to avoid this by structuring the paper as follows. The paper is subdivided into sections reflecting different "classes" of problems – learning from data, dealing with uncertainty, etc... – and for each section I have:

(i) presented briefly the appropriate framework;

(ii) introduced a general insurance application, without solving it, so that it is possible to follow the rest of the section with a concrete application in mind;

(iii) described the main available techniques to a reasonable level of detail;

(iv) analysed the application introduced in (ii) in terms of these techniques, developing where possible a very simple numerical example that hopefully contains the main ingredients of real-world applications;

(v) compared different techniques, where applicable.

Apart from some passing references, I have tried to avoid using examples not related with general insurance when illustrating the techniques.

Section 2 is devoted to learning from data, probably the most pervasive activity that actuaries are engaged in and that in which they have become more sophisticated. Section 3 is about dealing with uncertainty and expert knowledge – another ubiquitous problem for actuaries. Section 4 is about dealing with information that changes with time – as all information on risk always does. Section 5 is about making decision in an uncertain environment – for example decisions on prices, reserves, capital. Ultimately, making decisions is exactly why risk professionals need to understand risk! Section 6 deals with the collective behaviour of markets and with the interaction among different risk agents.

Chapter 2

Making data-based predictions (”learning from data”)

Did you say I’ve got a lot to learn?
Well don’t think I’m trying not to learn,
Since this is the perfect spot to learn
Go on, teach me tonight!
Gene De Paul, Sammy Cahn (1953): “Teach me tonight”

The past, the actuarial mantra goes, is not necessarily a good guide to the future. However, it is one of a few available in town and you decide to venture out without its indications at your peril. Many actuarial activities, such as experience rating and reserving, do at their core just that – using past data to create models about future behaviour. A set of techniques, deceptively denoted as *predictive modelling*, has also become widespread among actuaries. Predictive modelling attempts to infer from the data the factors that better explain the risk in order to price different policyholders a different amount of money, or to reserve different types of claims differently.

Experience rating and experience-based reserving or capital modelling, and even more so predictive modelling, are activities that the computational intelligence community would consider as examples of ”learning from data”. This section claims that the appropriate framework for prediction is indeed machine learning (aka statistical learning).

We will see that machine learning has a very rigorous way of building models based on data, parametrising them and validating them. At a time when, for regulatory purposes, more and more emphasis is placed on model validation (see for example the consultation paper CP56), being exposed to the model validation protocols of machine learning should be useful for risk professionals.

Notice the subtle linguistic change: in classical statistics and in actuarial practice one *calculates* – or rather, *estimates* – the parameters or the structure of a model: in machine learning one *learns* them.

There are two types of learning activities:

- **Supervised learning**: that is, given data points X_i and outputs Y_j, infer the characteristics of the model f that allows to predict the outputs from the inputs: $Y = f(X)$. Supervised learning is also called "learning with a teacher", where the "teacher" is any mechanism that gives us feedback on the true values of the outputs Y during the training stage. An example from general insurance is rating factor selection.
- **Unsupervised learning**: that is, finding patterns in data without being sure whether the patterns are right or wrong, because no teacher gives us feedback. Examples in which this is useful are exploratory analysis (for example exploratory classification of territories/drivers) and data mining.

The main reference for this section is "The Elements of Statistical Learning" by Hastie, Tibshirani and Friedman [HTF01].

2.1 Supervised learning

There are two types of prediction tasks in supervised learning: regression (quantitative outputs) and classification (qualitative outputs). Both can be viewed as a task in functional optimisation: given variables X (inputs), Y (outputs) with joint probability distribution $\Pr(X, Y)$, find $f(X)$ to predict Y such that the functional $\mathrm{EPE}(f)$ (expected prediction error) is minimised:

$$\mathrm{EPE}(f) = \mathrm{E}(L(Y, f(X))) \tag{2.1}$$

where $L(Y, f(X))$ is the so-called loss function. Examples of loss functions for regression are the squared loss ($L(Y, f(X)) = (Y - f(X))^2)$) used in least squares regression and the log-likelihood loss $L(Y, f(X)) = -2 \log \Pr_{f(X)}(Y)$

used in maximum likelihood estimation (which can be used for both regression and classification).

In the rest of this paper we will mainly focus on the problem of regression rather than classification. Many of the things that can be said for regression apply however equally well to classification.

Before looking at different techniques, let us present a simple specific example from general insurance to make it easier to follow the subsequent sections, and go through some general considerations regarding how models should be selected and validated.

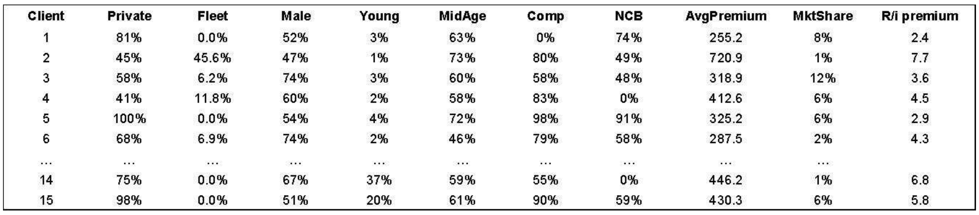
2.1.1 Example: Rating factor selection for reinsurance

Consider the problem of predicting the reinsurance premium charged for a given layer to different insurance companies for a specific line of business (motor, in this case) based on the profile of the insurer. This is an exercise in second-guessing how reinsurers rate excess-of-loss reinsurance for layers where loss experience is limited.

The factors that define the profile of the insurance company are chosen from a list including:

- % of private (v commercial) vehicles
- % of comprehensive (v TPFT) policies
- % of male drivers
- % of young ($\leq$ 25 y.o.) drivers
- % of mid-age (26-...) drivers
- % of drivers with $\geq$ 60% no claims discount
- Average direct (original) premium
- Market share
- ...

The input is in the form of a table such as that shown below. This table is based on real data but to make the companies unrecognisable data points have been scrambled and scaled to some degree.



Client	Private	Fleet	Male	Young	MidAge	Comp	NCB	AvgPremium	MktShare	R/i premium
1	81%	0.0%	52%	3%	63%	0%	74%	255.2	8%	2.4
2	45%	45.6%	47%	1%	73%	80%	49%	720.9	1%	7.7
3	58%	6.2%	74%	3%	60%	58%	48%	318.9	12%	3.6
4	41%	11.8%	60%	2%	58%	83%	0%	412.6	6%	4.5
5	100%	0.0%	54%	4%	72%	98%	91%	325.2	6%	2.9
6	68%	6.9%	74%	2%	46%	79%	58%	287.5	2%	4.3
...	...	...	...	...	...	...	...	...	...	
14	75%	0.0%	67%	37%	59%	55%	0%	446.2	1%	6.8
15	98%	0.0%	51%	20%	61%	90%	59%	430.3	6%	5.8

Figure 2.1: Reinsurance premium xs £5m and possible rating factors for different insurance companies. Figures are based on real data but are scrambled and randomised beyond recognition.

In the table shown in Figure 2.1, there are 15 observations. What we want is finding the features of the model that allow us to predict the reinsurance premium most efficiently. The features can be the factors themselves (% young, % male drivers, etc.) or a combination of them, for example (choosing a really odd example) "(%young)^2+log(%male)".

The feature selection problem can be formulated as follows: given a dictionary $\{\psi_\gamma\}_{\gamma\in\Gamma}$ of functions (each function is called a feature), select the features that are needed to represent the regression function, typically through a linear combination: $f_\beta(x) = \sum_{\gamma\in\Gamma}\beta_\gamma\psi_\gamma(x)$. Note that this formulation (borrowed from [DMDVR09]) is quite general and does not even require that a finite number of features be used.

2.1.2 Model selection and validation

When it comes to finding a good model, there are three main issues we are concerned about:

1. **Prediction accuracy.** When in machine learning we say that we want our model to have good predictive power, we mean it should fit a sample (the test sample) *which is independent from the one we used to parametrise the model* (that is, the training sample).

2. **Interpretation.** Our model should hopefully shed light on the underlying phenomenon, and should include only the most significant features. One can see this as a form of Occam's razor: given alternative theories with the same explanatory power, go for the simplest. One can

actually argue that ”understanding” always means producing compact versions of the data that we observe!

3. **Efficiency.** We should be able to determine what the best model is and what its parameters are with low computational complexity.

It turns out that the first two issues are intertwined. Figure 2.2, which gives the general behaviour of the expected prediction error as a function of model complexity, illustrates both issues at the same time.

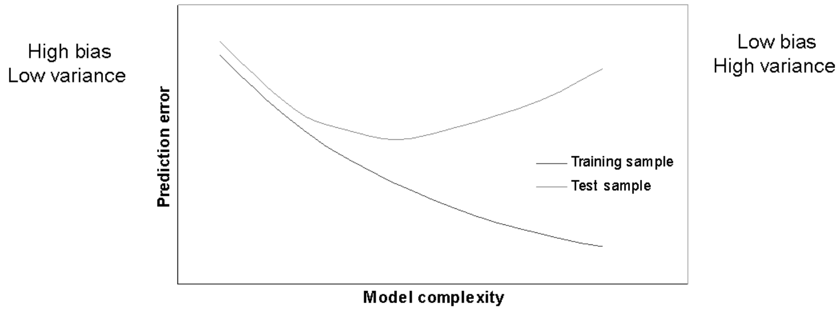


Figure 2.2: The trade-off between bias and variance. The right side of the graph corresponds to the region with low bias and high variance, whereas the left side of the graph corresponds to the region with high bias and low variance.

Model complexity can be measured in many different ways depending on the type of model. Often – for example, for a GLM exercise – it will be the number of parameters, but in other cases the measure will be less obvious (see for example the nearest neighbour example below). What Figure 2.2 is telling us is that the more complex the model becomes, the better the model will fit the training sample: eventually, we expect a perfect fit when the number of parameters becomes equal to the number of data points.

This can also be expressed by saying that the *training error*, which is the average loss over the training sample:

$$\overline{\text{err}} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}(x_i)) \tag{2.2}$$

always decreases as the complexity of the model increases.

However, the prediction error on an independent sample (the *test error*), which is defined as

$$\text{Err} = \text{E}[L(Y, \hat{f}(X))] \tag{2.3}$$

tells a different story: initially the prediction error decreases as the model is refined, but it then picks up again. There is a point at which the prediction error is minimal – that is the optimal complexity of the model.

What is happening is that there is a trade-off between bias and variance. The following decomposition holds when $Y = f(X) + \epsilon$, $E(\epsilon) = 0$, $Var(\epsilon) = \sigma_\epsilon^2$:

$$\begin{aligned} \text{Err}(x_0) &= E[(Y - \hat{f}(x_0))^2 | X = x_0] = \\ &= \sigma_\epsilon^2 + (E(\hat{f}(x_0)) - f(x_0))^2 + E[(\hat{f}(x_0) - E(\hat{f}(x_0))^2] = \\ &= \sigma_\epsilon^2 + \text{Bias}^2(\hat{f}(x_0)) + \text{Var}(\hat{f}(x_0)) \qquad (2.4) \end{aligned}$$

The first term is called the irreducible error – what we would normally refer to as process variance – something that depends on the inherent randomness of the underlying phenomenon. As to the second and the third term, it is usually the case that more complex models are able to decrease the bias (by adapting to the idiosyncrasies of the training sample) while at the same time increasing the variance.

The meaning of this decomposition is better explained by an example. Consider again the problem of predicting the reinsurance premium for an insurance company described in Section 2.1.1. A very simple strategy – and one which is regularly used in one version or another by risk professionals – is taking the average of the k nearest peers (after some metric on the space of factors has been defined). Very often $k = 1$. This method is illustrated in Figure 2.3.

Note that the parameter k is inversely related to the model complexity. For example, if $k = 1$, the model has basically almost as many parameters as data points, and is therefore very complex.

For this nearest neighbour example, the bias/variance decomposition becomes

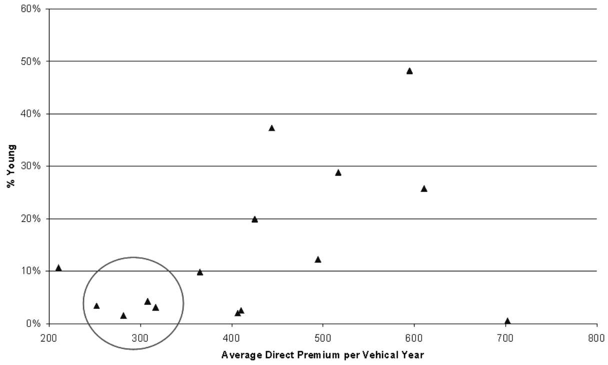


Figure 2.3: In this example, to estimate what reinsurance premium will be charged to an insurance company with average direct premium of £300 and a percentage of young drivers of 5%, the average of its closest peers (those inside the circle) is used.

$$\begin{aligned}\mathrm{Err}(x_0) &= E[(Y-\hat{f}(x_0))^2|X=x_0] = \\ &= \sigma_\epsilon^2 + \mathrm{Bias}^2(\hat{f}(x_0)) + \mathrm{Var}(\hat{f}(x_0)) = \\ &= \sigma_\epsilon^2 + [\frac{1}{k}\sum_{i=1}^{k} f(x_{(i)}) - f(x_0)]^2 + \frac{\sigma_\epsilon^2}{k} \qquad (2.5)\end{aligned}$$

For small values of k (high complexity), there is little bias, as only the most relevant peers are included. However, there are too few peers for an accurate average.

For high values of k (low complexity), a number of irrelevant peers are included and therefore the bias will increase, whereas the average will be more robust.

Model validation

So far we have discussed model selection based on the expected prediction error. The expected prediction error is also at the basis of the other goal we have, which is to *validate* the model. With Solvency II approaching, this is a topical issue in insurance. The reader is referred for example to the report of the working party on model validation and monitoring presented at GIRO 2009 by Berry et al. [BHMM09] for a review of state-of-the-art thinking in general insurance.

The first thing to remember when validating the model is that the only genuine way of validating the model is against data points that have not been used during the learning process.

Ideally one should divide the database randomly into three data sets:

Training set (say 50% of the original data set) to fit the model

Validation set (say 25%) to estimate prediction error for model selection purposes

Test set (25%) to estimate the prediction error of the final selected model!

This is a rigorous and general method, and has the advantage of fully decoupling model selection and model validation. To be rigorous, the test set should be used only once, during the final estimation of the prediction error.

When there is insufficient data, $\mathrm{EPE}(f)$ can be calculated approximately:

- By using k-fold cross-validation
- By using analytical methods such as AIC, BIC, MDL
- By using the bootstrap (randomised samples with replacement)

A key problem for the application to financial/actuarial problem is that *none of these methods can obviously assess the prediction error on new data from a changing/changed risk environment*!

We now look at each of these approximate methods in more detail. We are going to spend some time on this because as we will see later in the "experimental" section the way we calculate the expected prediction error *does* have an impact on model selection.

The treatment of these methods follows closely enough that of [HTF01] (Chapter 7), to which the reader seeking a greater wealth of information is referred to.

Cross-validation

Cross-validation, or more specifically K-fold cross-validation, estimates the extra-sample test error by dividing *at random* the data set into K different subsets. Each subset $k = 1, \dots K$ is in turn removed from the data set and the model is fitted to the remaining $K-1$ subsets. The subset k is used as a test set. The process is repeated for all subsets and the cross-validation estimate of the prediction error is given by

$$\mathrm{CV} = \frac{1}{N}\sum_{i=1}^{N} L(y_i, \hat{f}^{-\kappa(i)}(x_i)) \tag{2.6}$$

where $\kappa : \{1, \dots N\} \mapsto \{1, \dots K\}$ is an indexing function mapping the partition to which observation i is allocated by the randomisation, and $\hat{f}^{-k)}(x)$ is the fitted function calculated with the k-th subset removed.

For model selection purposes, we do not have a single model to cross-validate but a set of different models $f(x, \alpha)$indexed by a parameter α, which may represent for example the complexity of the model. For this set of models, we have a different value of $CV(\alpha)$ for every α, and *the function $CV(\alpha)$ is effectively an estimate of the test error curve* depicted in Figure 2.2. The optimal model is then $f(x, \hat{\alpha})$, where $\hat{\alpha}$ is the value that minimises $CV(\alpha)$.

One key issue with cross-validation is of course the choice of K. Anything from $K = 2$ to $K = N$ is possible. Perhaps unsurprisingly, the choice of K is also driven by a bias-variance tradeoff!

- The smaller K is, indeed, the larger is the bias introduced by cross-validation, as a large chunk of data is removed when we calculate the parameters on the training sets, and this means we have an inaccurate estimate of the parameters. This is probably better understood by thinking of the problem of estimating the parameters of a severity distribution, for example a Pareto distribution: the smaller the data set, the larger the parameter uncertainty and also the bias on the parameter – in the case of the Pareto, the bias is positive when calculated with MLE.

- The larger K is, however, the larger the variance. This is especially noticeable in the case $K = N$ (which removes a single point in turn and calculates the parameters of the model on the remaining $N-1$ points), which is usually referred to as the "leave-one-out strategy", and is suitable for cases where the data set is of limited size and removing a large percentage.

Overall, recommended values for K are usually $K = 5$ or $K = 10$. There's nothing special about 5 or 10 of course but they just happen to be values that everybody uses and therefore they're unlikely to raise eyebrows!

In all cases, cross-validation tends to overestimate to some degree the expected prediction error, because of the bias effect explained above.

Analytical criteria

Despite their simplicity, actuaries tend to have little acquaintance with model validation methods such as cross-validation or the three-sets protocol described at the beginning of this section on model selection and validation. This is changing (see the paper recently presented at GIRO by the model validation working party [BHMM09]). One reason for this relative lack of acquaintance is that most of the methods that actuaries use for feature selection come with built-in mechanisms to avoid using too many features.

For example, when deciding whether to include an extra-feature in a GLM exercise, one often applies at least a check on the standard error on the coefficient of the new factor: if that is of the same order of magnitude as the coefficient itself, this is an indication that the coefficient may not be significantly different from zero.

A quite standard approach in actuarial practice is to use the Akaike Information Criterion (AIC) (see below), by which every new degree of freedom can only be used if justified by the gain in the log-likelihood. However, several other analytical approaches are also possible. None of them requires validating the model on a different set.

Below we present some of these analytical methods. All of these methods punish complexity, playing the same role as regularisation. In the formulae below, d is the number of parameters, and N is the number of training points.

For a better understanding of how these methods work, notice that we can define three types of error to capture the "distance" between the sample and the model:

- The **training error** $\overline{\mathrm{err}}$, defined by Equation 2.2, giving the distance between the training sample and the model fitted on the same sample. This will obviously be biased downward.
- The test error Err, defined by Equation 2.3, giving the distance between the model and an independent sample. For the purpose of this discussion, we denote this as **extra-sample error**.

- The **in-sample error** $\mathrm{Err}_{\mathrm{in}}$, which gives the distance between the model and an independent sample *which shares with the original sample the points at which the model is evaluated.* The idea is that part of the discrepancy between the model and the test sample is due to the model being poor, and part is due to the fact that in general the points at which the model is calculated differ from those of the training sample. The in-sample error is defined as

$$\mathrm{Err}_{\mathrm{in}} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}_{\mathbf{y}}\mathrm{E}_{Y^{\mathrm{new}}} L(Y_i^{\mathrm{new}}, \hat{f}(x_i)) \tag{2.7}$$

  where the x_i's are the training points and Y^{new} are new responses at the same points.

All the methods below work by estimating the "optimism", i.e. the expected (usually positive) bias of the training error with respect to the in-sample error:

$$\mathrm{op} := \mathrm{Err}_{\mathrm{in}} - \mathrm{E}_{\mathbf{y}}(\overline{\mathrm{err}}) \tag{2.8}$$

and add it back to the training error $\overline{\mathrm{err}}$ to obtain an estimate of the in-sample error.

The question arises, of course: Why would we want to calculate the in-sample error, when what is really of relevance is the extra-sample error? The point is that although the in-sample error may not be a completely satisfactory estimate of model error, and therefore not completely satisfactory for model validation purposes, it usually does a good job for model selection purposes, where the main issue is not the absolute size of the error but how the error of one model compares to the other.

Since we will only be able to calculate the optimism approximately, the estimates of the in-sample error will have this form:

$$\hat{\mathrm{Err}}_{\mathrm{in}} = \overline{\mathrm{err}} + \hat{\mathrm{op}} \tag{2.9}$$

where as usual the hat symbol indicates an estimate.

We now have the tools to look into the different criteria.

Akaike Information Criterion (AIC)

The Akaike information is an estimate of Err_{in} that can be used in the case of a log-likelihood loss function (as is the case for GLM). It can be shown that the following holds:

$$\hat{\text{Err}}_{\text{in}} \propto AIC = -\frac{2}{N}\text{loglik} + 2\frac{d}{N} \tag{2.10}$$

asymptotically as $N \to \infty$. As usual, loglik in the equation above is the maximised log-likelihood.

The Akaike Information Criterion for model selection prescribes to choose the model with the smallest AIC over the set of models considered.

Bayesian Information Criterion (BIC)

The Bayesian Information Criterion has a similar form to AIC:

$$BIC = -2 \cdot \text{loglik} + d \cdot \log N \tag{2.11}$$

The BIC is proportional to the AIC but with the factor 2 replaced with $\log N$: as a consequence, when the number of data points is 8 or more the penalty for complex models is larger for BIC.

BIC arises by using a Bayesian approach to model selection, i.e. by choosing the model with the largest posterior distribution given the data (see [HTF01] and the discussion in Section 3.6). The good thing about BIC is that not only does it give a criterion for model selection but also allows to estimate the posterior probability of each model $\mathtt{M}_m$ as

$$\Pr(\mathtt{M}_m|Z) = \frac{e^{-\frac{1}{2}\cdot \text{BIC}_m}}{\sum_{l=1}^{M} e^{-\frac{1}{2}\cdot \text{BIC}_m}} \tag{2.12}$$

In the equation above, Z represents the training data.

Minimum Description Length (MDL)

The Minimum Description Length (MDL) gives a selection criterion which is formally identical to BIC, but which is motivated in terms of the theory of the compact representation of data and has its roots in Kolmogorov complexity. We cite it here although we do not expand on it because the underlying idea – that understanding reality (and therefore risk) is all about representing the data we have in the most compact form compatible with the information present in the data – is very important.

We refer the interested reader to [HTF01] for a short discussion, and to [GMP05] for a more extensive introduction to the subject.

Bootstrapping

The prediction error can also be estimated with the bootstrap. The main idea is to use bootstrap replicas of the data to refit the model and then calculate the error against the original training sample. Some modifications must be made to take into account that in this way the training sample and the different replica have observations in common and the expected prediction error would be underestimated. Again, the interested reader is referred to [HTF01] for a discussion.

Comparison of selection/validation methods

We now discuss how the different selection/validation methods compare to one another.

- AIC and BIC are less general than cross-validation. They can be used in their basic form only for the log-likelihood loss function and when the number of degrees of freedom is defined. For certain nonlinear models, such as neural networks, the number of degrees of freedom can sometimes be replaced with the effective number of parameters (see [HTF01]). However, for complex models this soon becomes very difficult and AIC, BIC become impractical;

- the AIC and BIC tend to overestimate the test error by a much larger amount than cross validation. This may be a problem in model selection (but only if this changes the relative order of models) and is certainly a problem in model validation, because the test error is exaggerated;

- BIC v AIC: BIC is asymptotically correct, in the sense that as the number N of data points tends to infinity, the true model is selected with probability 1. AIC does not have this property, and it tends to favour more complex models when N becomes very large. At the same time, BIC tends to over-punish complexity for finite samples and pick models that are too simple;

- BIC, AIC are simpler to calculate than cross-validation, as they don't require splitting the data into subsets and calculating the parameters many times.

Cross-validation therefore appears to be a far more general and more rigorous methodology, which however often requires more discipline and extra work and can therefore legitimately be replaced by analytical methods in many situations.

When we address regularisation, we will also see examples where cross-validation can be incorporated efficiently in the model selection algorithm.

Efficiency of selection/validation methods

The third element to model selection and validation is efficiency. Having a method that allows to select the best model to the best of our knowledge is not good enough: we also need to be able to do it within reasonable time constraints.

The relevance of this observation is that the brute-force approach to best subset selection, which simply looks all possible combinations of features to choose those to include in the model, is computationally intractable. This means that the time it takes to perform the selection increases exponentially with the number of features, and therefore rapidly becomes impossible when the number of features becomes large ("combinatorial explosion"). Note that combinatorial explosion is not a problem that can be solved by simply increasing the speed of computers: the gain in the size of the problems that can be solved will only increase logarithmically with processing speed. The reader interested to questions of computational complexity is referred to one of the classic books on the subject, that by Papadimitriou [Pap94].

How do we cope with the intractability of best subset selection, given that selecting the best subset of features is exactly what we have to do? There are two typical approaches to break intractability:

- Use **greedy algorithms**: that is, start with a very basic model (for example one that includes no features) and then start adding the features one by one, each time selecting the features that gives the most impressive results, e.g. the one that explains most of the variance. This is the approach which is probably the most familiar to actuaries. The problem with this approach is of course that there is no guarantee that a global minimum (a truly optimal subset of features) is reached: the order by which we add new features matters, and (especially when the features are correlated), choosing the feature explaining most of the variance may lead us astray. Consider the example in which we have three features X_1, X_2, X_3, and suppose that X_3 is actually (unbeknownst to us) $0.2 \cdot X_1 + 0.8 \cdot X_2$. Also suppose that the true model is $0.17 \cdot X_1 + 0.82 \cdot X_2$. Chances are that the greedy approach will select X_3 as the most important feature to be added, and since there will be noise in the observation it may find that adding any other of the variables will not need to any statistically significant reduction in variance. However, if we had tried all possible combinations: (X_1),(X_2), (X_3), (X_1, X_2), (X_1, X_3), (X_2, X_3), (X_1, X_2, X_3), we might have picked up the correct model despite the noise.

- An alternative to greedy algorithms are the **sparsity-based regularisation schemes** such as the lasso or elastic net, when will be introduced in Section 2.1.4. These work by adding a penalty term to the loss function, thus creating a regularised loss function. The size of the penalty term is controlled by a multiplicative parameter. By tuning these parameters, a different subset of features is automatically selected. This is called continuous subset selection and in many cases can be performed quite efficiently, as we will see in Section 2.1.4.

We now look into the most used techniques for feature selection, starting from that which is most popular among actuaries, generalised linear modelling. Regularisation theory and neural networks will also be described, and a comparison between the different methods will then be attempted.

2.1.3 Model selection and validation by GLM

GLM (Generalised Linear Modelling) is the recognised insurance industry standard for pricing personal lines (e.g. private motor, household) and small commercial lines. The main reference for GLM is the book by McCullagh and Nelder [MN90], but probably the most useful reference for actuaries is the Watson Wyatt tutorial by Duncan Anderson et al. A Practitioners Guide to Generalized Linear Models [MFS$^+$04], as it considers extensively the practical aspects.

GLM is an extension of the linear model. In the linear model, the dependent variables Y are related to the independent variables (or inputs) X through a matrix of coefficients β, with the addition of Gaussian noise ϵ which is usually assumed to be homoscedastic (independent of X):

$$Y = X \cdot \beta + \epsilon \tag{2.13}$$

where the components of Y are drawn from a Gaussian distribution (Gaussian noise):

$$Y_j \sim N(\mu_j, \sigma^2) \tag{2.14}$$

Generalised linear modelling extends this model into several directions:

- The linear combination of features is replaced with a linear combination of a wider (basically arbitrary) dictionary of functions $\psi_j(X)$, e.g. X_1, X_2, $X_1 \cdot X_2$, $\log(X_1 + X_2) \ldots$

$$\beta_1 \cdot X_1 + \ldots \beta_n \cdot X_n \mapsto \beta_1 \cdot \psi_1(X_1, \ldots X_n) + \ldots \beta_m \cdot \psi_m(X_1, \ldots X_n)$$

- The Gaussian noise is replaced by a more general type of noise, belonging to the so-called exponential family, which includes among the others Gaussian, binomial, Gamma, inverse Gaussian noise. The general form of the exponential family noise is:

$$f(y_i; \theta_i, \phi_i) = \exp\{\frac{\theta_i y_i - b(\theta_i)}{a_i(\phi)} + c(y_i, \phi)\}$$

- a further manipulation is achieved through the link function g, that links the linear combination above to the actual responses, yielding

$$Y = g^{-1}(\sum_j \beta_j \cdot \psi_j(X_1, \ldots X_n) + \epsilon) \tag{2.15}$$

  The link function is introduced so as to force certain behaviours on the responses: for example, if $g(x) = \log(x)$, the responses are constrained to be positive.

The expected value and the variance of Y_i are given by

$$E(Y_i) = \mu_i,\ Var(Y_i) = \phi \frac{Var(\mu_i)}{\omega_i} \tag{2.16}$$

which is more flexible than the homoscedastic requirement of linear models.

As an example, claims frequency is usually modelled by a GLM with a Poisson link function, with Poisson noise, and $V(\mu_i) = \mu_i$, $\phi = 1$, whereas the weights ω_i are usually chosen to be the exposures. Claims amount are often modelled by using Gamma noise. See [MFS$^+$04] for a more thorough discussion.

The parameters of a given model $f(X)$ can be calculated by maximum likelihood, or in other words by minimising a log-likelihood loss function:

$$L(Y, f(X)) = -2 \cdot \log \Pr_{f(X)}(Y) \tag{2.17}$$

Notice that the factor "2" has been introduced so that in the case of Gaussian noise this becomes the standard squared loss.

The model is usually selected using a greedy approach. This comes in two varieties, forward selection and backward selection. Forward selection starts from the simplest model, model $Y =$ constant, and adds at each step the function $\psi_j(X_1, \ldots X_n)$ that reduces the loss function by the largest degree. Backward selection starts from the most complex model (if this is defined) and removes functions one by one. We will normally adopt forward selection in the examples. In both cases, however, the goal is to break the computational intractability of best subset selection.

To decide whether the selected additional function in forward selection is actually an improvement, a typical test is to check whether the AIC (see Section 2.1.2) decreases or, more simply, whether the coefficient of the newly introduced function is significantly different from zero.

2.1.4 Model selection and validation by regularisation

The main idea of regularisation as a tool for model selection is that in order to minimise $\mathrm{EPE}(f) = \mathrm{E}(L(Y, f(X)))$ on a test sample you should minimise a different (regularised) functional:

$$\hat{f} = \mathrm{argmin}_\beta \{\mathrm{E}(L(Y, f(X))) + \lambda g_\beta(X)\} \tag{2.18}$$

on the training sample.

The term $g_\beta(X)$ is called a penalty term and embeds our prior knowledge on what the desirable features of the parameters are. The most famous example is ridge regression, in which $g_\beta(X) = ||\beta||^2_{l^2}$: in this case the idea is that coefficients should be as small as possible.

Although theoretically any loss function could be used in Equation 2.18, in practice quadratic loss is usually adopted when modelling quantitative (rather than categorical) responses, and most computational methods to derive solutions have been devised for the quadratic loss case.

The aim in all cases will therefore be to choose β so as to minimise the least squares distance between inputs and responses:

$$E_n(\beta) = ||Y - f_\beta(X)||^2_{l_2} \tag{2.19}$$

on an independent sample.

The norm $||\cdot||_{l_2}$ is defined as the usual sum of squared differences: if a, b are two vectors in $\Re^n$, then $||a - b||^2_{l_2} = (a_1 - b_1)^2 + \ldots + (a_n - b_n)^2$.

To achieve this, we have seen that the best strategy is not to try to minimise the quadratic loss on the training sample. Rather, a regularised quadratic loss may have to be used, aiming to keep the complexity of the solution under control.

In the following we look at different types of regression.

Non-regularised regression

The "vanilla" case is of course the non-regularised regression of Equation 2.19. The solution in this case is given by the standard least squares solution:

$$\hat{\beta} = (X^T X)^{-1} X^T y \tag{2.20}$$

Note how the solution is a linear function of the observations $\underline{y}$.

Ridge regression

The ridge regression uses an l_2 penalty which punishes large coefficients:

$$E_n^\lambda(\beta) = ||Y - f_\beta(x)||_{l_2}^2 + \lambda||\beta||_{l_2}^2 \tag{2.21}$$

The reasoning behind this is that in the case of correlated variables, solutions where one large coefficient is compensated by another large coefficient of opposite sign for a correlated variable are common. The l_2 penalty tends to shrink the coefficients, keeping this behaviour at bay, and therefore preventing the appearance of spurious information.

The solution is again a simple linear function of the observations y:

$$\hat{\beta} = (X^T X + \lambda I)^{-1} X^T y \tag{2.22}$$

The lasso penalty

This was introduced by [Tibshirani, 1996]. As with the ridge regression, it imposes a penalty on the size of the regression coefficients, but through an l_1 penalty term:

$$E_n^\mu(\beta) = ||Y - f_\beta(x)||_{l_2}^2 + \mu||\beta||_{l_1} \tag{2.23}$$

This penalty term has the (perhaps surprising) effect of forcing some of the coefficients to be zero, operating what is effectively a continuous subsect selection and enforcing *sparsity.*

No nice analytical solution can be provided in this case. The solution can however be obtained by quadratic programming. The reason for this is perhaps clearer by noticing that the problem can be framed as a constrained optimisation problem:

$$\text{Minimise } \|Y - f_\beta(X)\|_{l_2}^2 \text{ subject to } \|\beta\|_{l_1} < t$$

An alternative (and more efficient) method to quadratic programming is given in [EHJT04]) and is based on the homotopy method of Osborne, Presnell Turlach [OPT98].

Least Angle Regression (”LARS”) includes the lasso as a special case. LARS with a lasso implementation can be used to calculate the lasso estimates for a problem for *all* values of the constraint, and does that very efficiently (same order of magnitude as a standard least squares method).

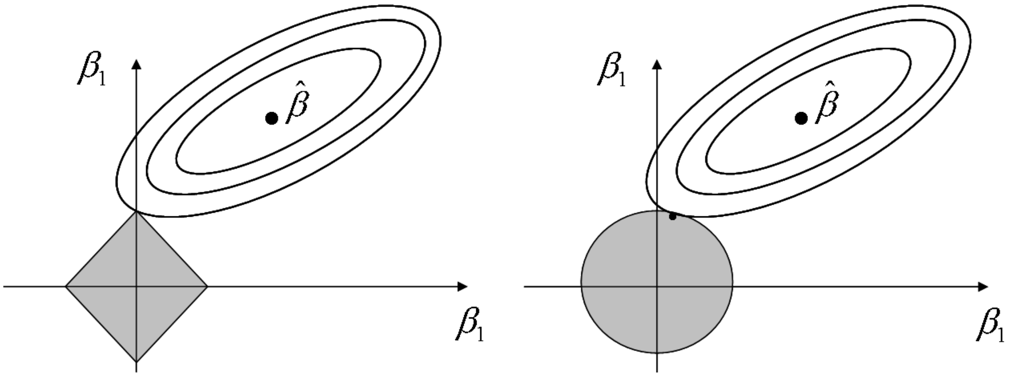


Figure 2.4: In the figure above, the ellipses represent the quadratic loss function and the geometric figure around the (0,0) point represents the constraint (left: lasso; right: ridge regression). The solution of the problem must lie at the intersection between the loss function and the constraint. This intersection is usually at one of the vertices for the lasso (unless the loss function happens to be tangential to the constraint).

Other, even more efficient methods using efficient path descent are described in [Has].

The obvious advantage of the lasso is that it operates simultaneously the shrinkage of coefficients and the selection of the relevant variables without guidance by the analyst.

The most significant drawbacks of the lasso are [ZH05]:

- in the case where the number of predictors p is larger than the number N of observations (the so-called "large p, small N scenario" this may seem a far-fetched situation, but it is quite common in certain applications, such as microarray data analysis), the lasso selects at most n variables before it saturates, because of the nature of the convex optimization problem;
- in the usual case where $N > p$, if there are high correlations between the predictors the prediction performance of the lasso has been empirically observed [Tib96] to be worse than that of ridge regression;
- if there is a group of variables among which the pairwise correlations are very high, the lasso tends to select only one variable from the group.

Note that the method can be extended to employ an l_p-penalty.

Applications to general insurance.

As we shall see later in a specific example, the lasso promises to provide a simple alternative way to forward/backward selection used in GLM to operate feature selection. This way is both rigorous and efficient. As to the drawbacks listed above, the “large p, small N scenario” is usually not a concern, at least for personal line insurance applications, but it may become so in other applications (see Section 2.1.4 below). The deterioration of prediction performance in the $N > p$ scenario when highly correlated variables are present is instead something that is quite applicable in our context, as many variables that are considered during the underwriting process, such as the age of the driver and whether the policy purchased is comprehensive or TPFT (third-party, fire and theft) liability only are strongly correlated. The presence of small networks of predictors that need to be preserved, however, seems to be quite specific to biology: there seems no harm in selecting any of a group of correlated variables as long as the overall performance is good.

Lasso in practice.

A simple implementation of the lasso is included in the ”lars” R package by Trevor Hastie. The package includes plotting facilities and the calculation of the cross-validation estimate of the expected prediction error for all values of the regularisation parameter.

Elastic net

In some contexts, such as microarray data analysis, the number of predictors (which in this case are genes) is typically far larger than the number of observations; also, some of the predictors are highly correlated and when looking for a model they should all be part of the solution as they reflect the fact that genes tend to be part of small interacting networks. These networks must be identified fully in order to understand the underlying biological mechanisms (see for example [MMTV]).

In these contexts the lasso is not adequate (see the list of drawbacks in the previous section). This has provided the motivation for elastic net regularisation, a modification of the lasso introduced by Zou and Hastie [ZH05] in 2005.

Zou and Hastie proposed the use of a composite penalty, with an l_1 and an l_2 term:

$$E_n^{\lambda,\mu}(\beta) = ||Y - f_\beta(x)||_{l_2}^2 + \lambda||\beta||_{l_2}^2 + \mu||\beta||_{l_1} \tag{2.24}$$

The first term ensures that only a few relevant features are selected, whereas the second term tries to avoid the excesses of the lasso, ensuring that no element in a group of correlated variables is picked out at random excluding the others. Note that unlike other methods (the interested reader is referred to citations [31, 42, 22] in De Vito's paper) the groups of correlated variables need not be known in advance.

De Mol et al. [DMDVR09] proved that under the assumption that the regression function admits a sparse representation on the dictionary, there exists a particular elastic-net representation of the regression function such that, if the number of data increases, the elastic-net estimator is consistent not only for prediction but also for variable/feature selection. They also provided a different interpretation of the elastic net solution in terms of the fixed point of a contraction in a Banach space, and a different algorithm for finding this solution.

Applications to general insurance.

As we have commented above, in personal lines insurance – where rating factors analysis has started being used, and which is currently the main application for it – we are in a "small p, large N" scenario, with the number of predictors in the order of 100 and the number of observations at least 100 times larger. However, in other contexts – such as commercial lines insurance, or reinsurance – we might sometimes be able to collect a number of predictors that exceed the number of observations, and using the elastic net should be considered.

Also, there is often a high correlation among variables, although the case for preserving "cliques" at all costs is weaker.

In any case, it is certainly worth in these cases trying to analyse the elastic net alongside the lasso and see if the prediction performance improves.

Regularised regression in the Bayesian framework

It is interesting to notice that regularised regression can be looked at in the framework of Bayesian analysis. In this framework, we are looking for the values of the parameters that maximise the posterior probability given the data, and the penalty term corresponds to the prior probability on the coefficients.

For example,

- the ridge regression penalty term corresponds to a multivariate normal prior probability centred around zero: $\Pr(\beta) \propto \exp(-\lambda ||\beta||_{l_2}^2)$
- the lasso regression, on the other hand, corresponds to a Laplace distribution: $\Pr(\beta) \propto \exp(-\lambda ||\beta||_{l_1})$;
- the elastic net corresponds to the distribution $\Pr(\beta) \propto \exp(-\lambda ||\beta||_{l_2}^2 - \lambda' ||\beta||_{l_1})$ (see [LL09]).

The reason why this is important is that it sheds further light on the nature of regularisation, which is ultimately the incorporation of prior knowledge of the model into the fitting process. The ridge regression reflects the belief that the parameters are small; the lasso reflects the belief that the model is sparse; the elastic net reflects both beliefs.

2.1.5 Model selection and validation by neural networks

Neural networks were initially developed as simple models for the human brain. The basic units of the network correspond to the neurons, and the connections between these units correspond to the synapses.

This provided the inspiration for much research in artificial intelligence, in the hope to emulate the following characteristics of the brain (see [HKP91]):

- It is robust and fault tolerant
- It is flexible and able to learn
- It can deal with information that is fuzzy, probabilistic, noisy, or inconsistent
- It is highly parallel
- It is small, compact, and dissipates very little power

Research on artificial neural networks can be traced back to a work by McCulloch and Pitts [MP43]. The first wave of research has culminated in a number of works by Rosenblatt et al. [Ros58] in the 1960s on perceptrons, which were networks of neurons organised in layers, such as that in Figure

2.5, with weights associated with each connection. Rosenblatt initially focused on single-layer perceptrons (with no hidden layers between inputs and responses) and was able to prove how the weights of a neural network can be updated so as to guarantee the convergence of a learning algorithm to perform a desired computation. However, Minsky and Papert, in their book Perceptrons [MP69], proved that some elementary computations such as the exclusive-or (XOR) cannot be performed by Rosenblatt's one-layer perceptrons. Rosenblatt suspected that by introducing hidden layers (as in Figure 2.5, where there is one hidden layer) this problem could be overcome, but no learning algorithms had yet been discovered for this case.

This proved a powerful setback for neural networks: different approaches to AI were thought to be more promising by the AI community, which abandoned the neural network paradigm for almost 20 years. Research stagnated until in the mid-80s the back-propagation algorithm for updating the weights of a neural network was discovered (or rather re-discovered) by Rumelhart, Hinton, and Williams [RHM87]. Neural networks have ever since been applied to a wide range of problems, from computer vision to particle physics, and as John Denker, an aviation scientist, once famously remarked, they seem to provide "the second best way of doing just about anything". Many researchers, however, have found it difficult to accept this paradigm, as the solutions it provides are not transparent – neural networks work, but it is difficult to understand why!

A more thorough story of how neural networks developed, and went from obscurity to hype to obscurity again and then to wider acceptance, is outlined in the book by Hertz, Kroch and Palmer [HKP91], from which the highlights above have been taken.

A singularly balanced view on neural networks is expressed in [HTF01]. Rejecting the hype and mystery surrounding them and also avoiding an outright rejection, the authors notice that neural networks are nothing but non-linear statistical models, much like project pursuit regression [HTF01].

Neural networks come in many flavours, the most classical of which is the single hidden layer, feed-forward neural network shown in Figure 2.5. The single hidden layer, feed-forward neural network shown there can be used for both regression and classification, although for regression there is only one output (Y) whereas for classification more outputs are needed, the number K of units at the top being equal to the number of classes.

The units at the bottom layer are called either features or simply inputs; those in the hidden layer are called hidden units or derived features; and

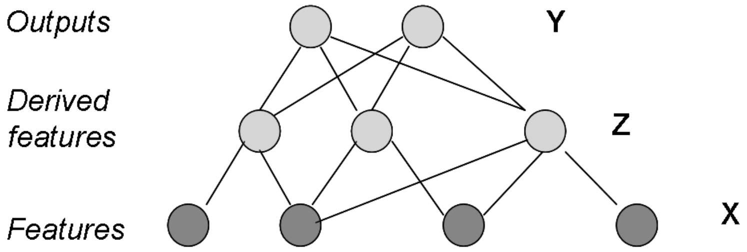


Figure 2.5: A single hidden layer, feed-forward neural network.

those of the top layer are called either outputs or responses.

Let us outline the main ingredients of the neural network model of Figure 2.5 and the equations regulating its behaviour:

- Inputs: $X_1, \ldots X_p$
- Outputs: $Y_1, \ldots Y_K$
- Derived features: $Z_1, \ldots Z_M$
- $Z_m = \sigma(\alpha_{0m} + \alpha_m^T X)$, $m = 1, \ldots M$, where σ is the activation fuction. The most popular choice is the sigmoid function, $\sigma(v) = (1 + \exp(-v))^{-1}$
- $T_k = \beta_{0k} + \beta_k^T Z$, $k = 1, \ldots K$
- $Y_k = f_k(X) = g_k(T)$, $k = 1, \ldots K$. Typically, $g(T) = T$ (identity function) for regression, and $g_k(T) = \frac{e^{T_k}}{\sum_{l=1}^K e^{T_l}}$ (softmax function) for classification.

Note that in the early experiments on neural computation the step function was used instead of the sigmoid activation – however, it was later found that the sigmoid function gave more flexibility for optimisation. Alternatively, Gaussian radial basis functions can be used.

Solving a problem with a neural network means finding the correct paramters (here called *weights*) to use in a given situation. The complete set of weights θ consists of

$$\theta = \{\alpha_{0m}, \alpha_m; m = 1, 2, \dots M\} \cup \{\beta_{0k}, \beta_k; k = 1, 2, \dots K\}$$

The parameters are learned from the data during the training stage, by minimising one of the following loss functions: sum-of-squares

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} (y_{ik} - f_k(x_i))^2 \tag{2.25}$$

for regression and cross-entropy

$$R(\theta) = \sum_{k=1}^{K} \sum_{i=1}^{N} y_{ik} \log f_k(x_i) \tag{2.26}$$

for classification.

The minimisation can be performed for example by back-propagation, which is an application of the gradient descent method.

A key issue with neural networks is that since there are usually many weights there is a danger of overfitting, by adapting too closely to the nuances of the data. To avoid this one can introduce a penalty term and find a regularised solution, for example by minimising $R(\theta) + \lambda J(\theta)$ instead of $R(\theta)$, where $J(\theta) = \sum_{km} \beta_{km}^2 + \sum_{ml} \alpha_{ml}^2$. This form of regularisation is called *weights decay* and is analogous to ridge regression. Alternatively, one can use (as it used to be in the early days of neural networks) an early stopping rule, by which the training of the model is stopped way before a global minimum is reached.

Importantly, it should be noted that neural networks are extremely general and that when they include hidden layers they can learn any function (see the list of examples in [LP96]). Furthermore, they can be validated like any other method and should not therefore be rejected on the grounds that their results are not reliable.

There is, however, another important reason why neural networks are not suitable for many applications, and it is their lack of transparency: neural networks provide, in the words of [HTF01], **prediction without interpretation**. In other words, they provide an answer (for example how much a specific customer should be charged) without explaining why.

Finally, neural networks tend to require more computational resources to produce a fitted model.

Applications to insurance

In certain circumstances, the opacity of neural networks may be an advantage as a detailed model specification is not required.

However, the fact that neural networks provide "prediction without interpretation" is certainly a problem in general because they are not going to help us *understanding* risk. It is therefore difficult to see how underwriters, managers at an insurance company or the regulator could accept lightly pricing, reserving, capital modelling decisions based on neural networks.

Despite this problem, some applications of neural networks to general insurance have been attempted, for example by Dugas et al. [DBC$^+$03], which investigated the use of neural networks for motor rating in North America.

A comparison of neural networks and GLMs was attempted in a paper by Lowe and Pryor [LP96] in 1996. The main message of this paper was consistent with what stated above: neural networks have greater generality with respect to GLM thanks to the non-linearities they allow, and they can also spare the need and time of detailed modelling; however, the opacity and the computational demands of neural networks may prevent their wider applications to general insurance.

My view is that despite the fact that neural networks are a sound and very general non-linear statistical model which can be validated against an independent sample like any other model, the problem of "prediction without interpretation" is the overwhelming consideration in a world that has to make justified business decision. Unless the neural networks community succeeds in its quest to provide interpretable results (some authors have for example suggested performing principal components analysis on the weights for this purpose), the scope of neural networks' applicability to insurance will be limited. I would certainly not advocate their use.

On the other hand, I think that neural networks could be used "behind the scenes" to provide a benchmark against which to judge other models. If for example GLMs or regularised regression produce solutions that have a test error far higher than that provided by neural modelling, that might be interpreted a sign that our models are not capturing the right features of the model and that we should try harder. Also, there is certainly no reason why not use neural networks for exploratory analysis, as is currently done in many data mining applications.

In the following, neural networks will not be explored further (except for an overall comparison among different learning methods in Section 2.1.8).

2.1.6 Practical general insurance example: Rating factors for reinsurance pricing

The first general insurance example is that already outlined in Section 2.1.1. This is not a very sophisticated example and a multivariate Gaussian model of the linear variety will suffice: *i.e.*, we model the reinsurance premium Y as a linear function of the rating factors:

$$Y = a_0 + a_1 X_1 + \ldots + a_n X_n \tag{2.27}$$

GLM approach.

The standard approach to this problem with GLM is through forward or backward selection of the best model. Forward selection is a form of greedy approach and starts with the simplest model, then adding at each step the variable which reduces the variance between the model and the empirical data by the largest amount, until the addition of a new variable does not add a statistically significant advantage. Backward selection (which will not be used here) works the other way round, starting from the complete model (that where all the variables are included) and removing one variable at a time.

In our case, we start from the model $Y = a_0$ (also indicated as $Y \sim 1$), which has variance 54.8 (referred to as null deviance).

In the first step, we add a single rating factor. We try all of them in turn, and for each of them we calculate the residual variance. We select the model with the lowest residual variance, and we check that the reduction in variance is enough to introduce a new degree of freedom.

It turns out that the best factor is "Average Direct Premium". This leads

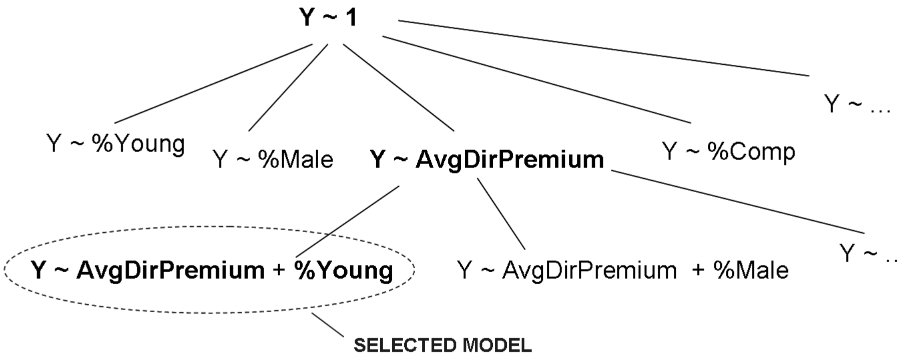


Figure 2.6: Illustration of the forward selection method.

us to the model $Y = a_0 + a_{\text{ADP}) \cdot \text{ADP}}$ ($Y \sim$ ADP), which has a variance of 12.1. To understand whether this is actually a significant gain, we consider two tests.

(i) One (informal) is based on the standard error on the coefficients. A coefficient which has a standard error too close to the actual value of the coefficient is drowned in uncertainty and cannot be trusted. In our case, the coefficient value is[1] $a_{\text{ADP}} = 0.021$ with a standard error of $\text{se}(a_{\text{ADP}}) = 0.003$. This corresponds to a t-value of about 7. The probability of having this value of t by chance (that is, the probability that the true value of the coefficient a_{ADP} is actually 0) is about $Pr(> |t|) = 6 \cdot 10^{-6}$. The coefficient is therefore highly significant;

(ii) A more formal test is the Akaike Information Criterion (see Section 2.1.2), which can be replaced with any of a number of similar tests, each with their own pros and cons. The AIC value for the model $Y = a_0 + a_{\text{ADP}) \cdot \text{ADP}}$ is 46.9. Since this is less than that for the model $Y = a_0$ (AIC=69.1), the addition of the variable ADP is accepted.

We now try to refine the model adding a second coefficient. It turns out that the best model is now $Y = a_0 + a_{\text{ADP}}\text{ADP} + a_Y\%\text{Young}$, and that the addition of %Young still leads to a significant statistical gain (see Table **??**).

If we now try to add a third factor, we find that even the factor that gives the largest reduction in the residual variance (MktShare, representing the market share of the insurance company), does not give any statistically significant gain (see again Table 2.1). Therefore our final model according to our greedy

[1] Please note that since this experiment is based on real data from 2008 renewal prices, the coefficients have been scaled by a random factor.

Model	Variance	Pr(> $\|t\|$)	AIC
$Y \sim 1$	54.8	$2 \cdot 10^{-8}$	69.1
$Y \sim$ ADP	12.1	$6 \cdot 10^{-6}$	46.9
$Y \sim$ ADP + %Young	7.1	0.009	40.3
$Y \sim$ ADP + %Young + MktShare	6.6	0.39	41.3

Table 2.1: Increasingly complex models for the reinsurance premium, and their statistical significance. Every time the variable that explains most of the residual variance is added to the model, the variance is reduced. The column $Pr(> |t|)$ gives the probability that the ratio between the coefficient and the standard error *of the latest variable added* is larger than $|t|$ by chance although the true value of the coefficient is zero. A value above a significance value of 5-10% indicates that the coefficient is not significantly different from zero. More rigorously, the drop in variance obtained by the addition of a new variable is statistically significant only if the value of AIC (fourth column) decreases. Both these criteria indicate in this case that the best model is $Y \sim$ ADP + %Young.

forward selection approach is $Y = a_0 + a_{\mathrm{ADP}}\mathrm{ADP} + a_Y\%\mathrm{Young}$.

The main problem with this approach lies in the model selection process, which does not guarantee that it will not fall into local minima, and is relatively inefficient. Notice that automatic selection is usually available at least for the multivariate Gaussian model – e.g., in R there is a function called stepAIC() that does automatic selection, but only for Gaussian models.

Regularisation approach.

We now use a simple lasso regularisation approach to solve the same problem. We use an R package called "LARS", authored by Hastie (2007). This package performs an automatic selection process which is guaranteed to find the global minimum, and it does it within the time it takes to do a single least square regression.

The coefficient values are plotted against the regularisation parameter λ, which is expressed as a fraction of $t^* = \sum_1^p |\hat{\beta}_j|$, where $\hat{\beta}_j$ is the least squares estimate of coefficient β_j. Note that when the regularisation parameter becomes larger than t^*, the lasso estimates coincide with those of least squares regression. Owing to the nature of the lasso, the coefficients are all zeros when $\lambda = 0$, then as we increase λ the coefficients become positive one by one, as in Figure 2.7.

As figure 2.7 shows, the first coefficient to become non-zero is that for the

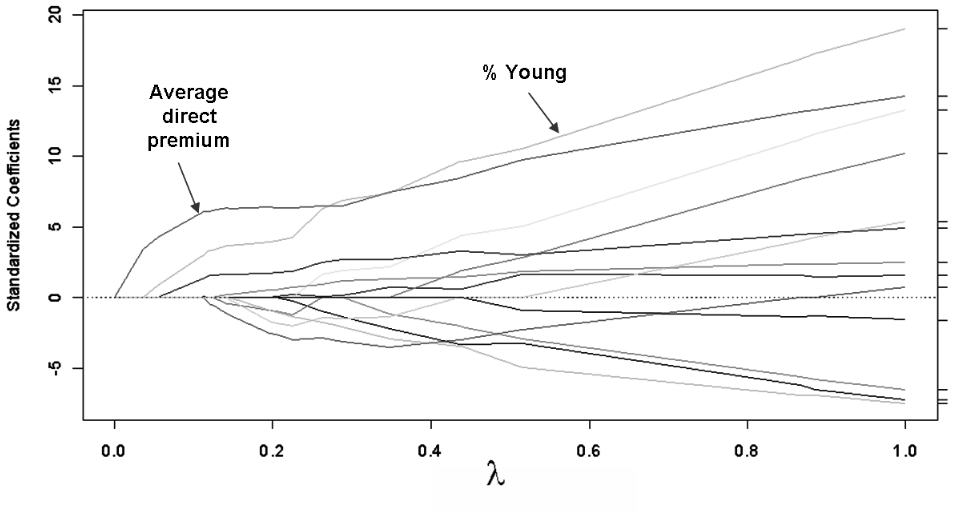


Figure 2.7: Illustration of the least-angle regression method (LARS) to generate all lasso solutions for different values of the regularisation coefficient.

average direct premium, and the second is that for the percentage of young drivers, then all the others, one by one.

The same package offers a method to decide which value of λ should be selected based on cross-validation, as shown in figure 2.8, which gives the expected prediction error as a function of the value of λ, again expressed as a fraction to the largest coefficient.

The minimum value of the prediction error is obtained around 0.05. This corresponds to including the variables ADP and %Young, exactly as for the GLM model. The model has however been obtained effortlessly and is guaranteed to yield the global minimum.

2.1.7 Practical general insurance example: Modelling claim frequency for personal or commercial line business

In the example in Section 2.1.6 the underlying noise model (or 'error structure') was assumed to be Gaussian, and the link function was assumed to be the identity function. This suited the application we were considering. How-

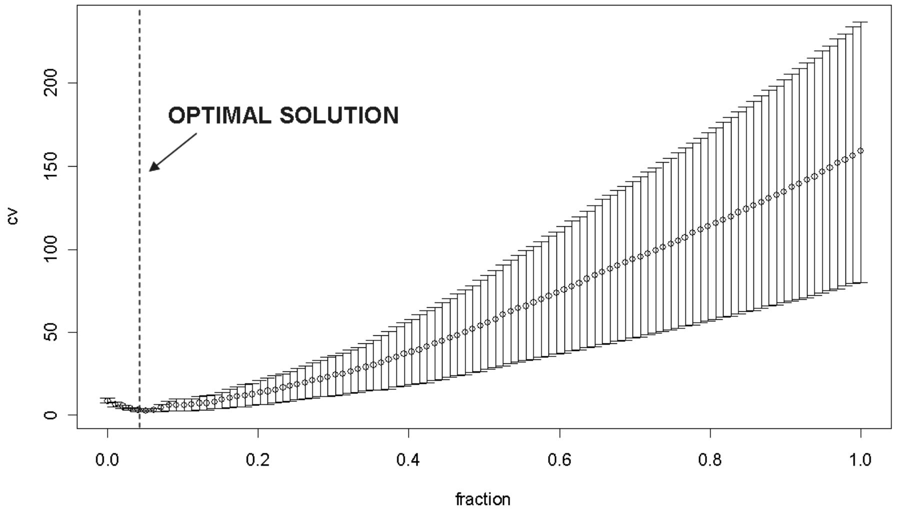


Figure 2.8: Expected prediction error for different values of the regularisation parameter, obtained through cross validation.

ever, in many circumstances a Gaussian noise structure is not good enough. In common GLM applications, for example, we are looking to parametrise a frequency and a severity model separately, and a common assumption for those is to use a Poisson and a Gamma noise model respectively (see, e.g., the Watson Wyatt's report **??**). As many regularisation algorithms assume a quadratic loss, the effect of using a quadratic loss to determine parameters for models with a non-Gaussian noise model should be investigated. Although there is no mandatory connection between the noise model and the loss function, the quadratic loss is quite a natural choice in connection with Gaussian model, as the logP loss function and the quadratic loss are coincident for Gaussian noise.

In this section we will therefore perform a controlled experiment (i.e., one with artificial data coming from a model of which we know everything) using a Poisson noise model, comparing the results obtained with the GLM and the lasso approach.

The model predicts the number of claims for different categories of customers/clients. It has the form $Y \sim \mathrm{Poi}(E \cdot \nu)$, where E is the overall exposure (e.g. number of accounts/customers) and ν is given by a simple linear relationship:

Factor	No of levels	Level 1	Level 2	Level 3	Level 4
Sex	2	42%	58%	0%	0%
Age	4	25%	30%	30%	15%
Region	3	30%	40%	30%	0%
Profession	2	20%	60%	20%	0%
NCB	4	30%	40%	20%	10%

Table 2.2: Exposure percentages for the different factors

$$\nu = \exp(-3 - 0.3 \cdot \text{Age} + 0.2 \cdot \text{Sex} + 0.15 \cdot \text{Region} - 0.4 \cdot \text{NCB} + 0.1 \cdot \text{Profession}) \tag{2.28}$$

where Age, Sex, Region, NCB, Profession are discrete variables with 4, 2, 3, 4, 2 levels respectively. The following variables are also introduced, which have no bearing on the frequency: Colour, Garden, Dumb1, Dumb2, Dumb3, all binary.

The portfolio composition in terms of the variables above is defined by the table below. For simplicity, we assume that all exposures are independent, so that to obtain the exposure for a given combination one simply multiplies the overall exposure by the percentage of each variable (including the dummy ones) in that level:

$$E(\text{Age} = l, \text{Sex} = m \ldots) = \text{E(all)Perc}(\text{Age} = l)\text{Perc}(\text{Sex} = m) \ldots$$

.

The exposure percentages are given in Table 2.2.

A data set of 1,000 (not necessarily all different) random combinations of levels of the ten variables above is generated. For each combination γ:

1. the exposure E_γ is calculated according to the prescription above;
2. the theoretical Poisson rate ν_γ per unit of exposure is calculated according to Equation 2.28;
3. this is multiplied by the calculated exposure to obtain the theoretical absolute frequency $E_\gamma \nu_\gamma$;
4. a random number n_γ is then drawn from a Poisson distribution with that frequency: $n_\gamma \sim (E_\gamma \nu_\gamma)$.

At this point, the true model is forgotten and an exercise in model selection/parametrisation/validation is run for both GLM and regularisation, according to the methodology explained in the previous section.

In the case of GLM, it is assumed that the noise model is Poisson and that the link function is logarithmic – which is exactly the case in our artificial set-up.

In the case of lasso regularisation, we first get rid of the entries for which the claim count is zero and then we take the logarithm of the responses. A linear fit $\log Y = a_0 + \sum_j a_j X_j$ is then attempted using the squared loss.

Both GLM and lasso regularisation are run for three different overall exposure levels: $E_1 = 10,000,000$, $E_2 = 1,000,000$, $E_3 = 100,000$. These correspond to an average Poisson rate across categories of about (respectively) 300, 30 and 3. The reason for this range is that we want to see where the difference between the Poisson and the Gaussian behaviour really emerges, i.e. for small values of the Poisson rate.

GLM gives very good results across all values of overall exposure if it is allowed by the model selection algorithm to go up to the true number of variables: the coefficients are very close to their true value. However, if one uses one of the analytical criteria described in Section 2.1.2, one usually has to stop way before getting to the right number of variables. By using, for example, the t-value for the standard errors of the coefficients as a guide to selecting the ones that are significantly different from zero, we find that the most complex model allowed is $\log(Y) = a_0 + a_1 \cdot \text{NCB}$. By using AIC as a guiding principle, we find we have to stop at $\log(Y) = a_0 + a_1 \cdot \text{NCB} + a_2 \cdot \text{Age}$ (see Table 2.3). This confirms that AIC, BIC etc. tend to overstate the prediction error and are therefore unduly restrictive. Using cross-validation is better in this case; however, cross-validation tends to understate the prediction error.

Regularised lasso performance depends more critically on the overall exposure level: the performance degrades as the average Poisson rate decreases. This is perhaps not surprising since we are using a squared loss function rather than a log P loss function. However, the results are still very good when the

Model	Exp=100k	Exp=1m	Exp=10m
$Y \sim 1$	30.5	12.1	10.0
$Y \sim$ NCB	24.7	6.8	4.3
$Y \sim$ NCB + Age	21.3	3.8	1.2
$Y \sim$ NCB + Age + Region	20.7	3.2	0.7

Table 2.3: Forward selection algorithm for GLM using the Akaike Information Criterion, for different levels of exposure (100k to 10m). New variables are added until the reduction in frequency is statistically not significant anymore. In the case above, the winning model is $Y \sim \text{NCB} + \text{Age}$ for all levels of exposure, as the reduction in variance in the last row is less than 2 (twice the number of degrees of freedom added).

average Poisson rate is around 30.

With these simulated data, we actually get the unusual effect that the expected prediction error flattens at around 0.8 and does not pick up significantly after that. This amounts to choosing a model with an excessive number of variables. However, it should be noticed that the values of the coefficients of the dumb variables (Colour, Dumb1, Dumb2...) are actually very small; therefore, the inclusion of these variables is only notional: unclean, but harmless.

In order to show the degradation effect for the regularised lasso, Table 2.4 shows the coefficients of the selected lasso model for different values of the exposure. As it can be seen, the difference between GLM and regularised regression is not large when the exposure is 10m or 1m. However, the performance of regularised regression is not good for very low levels of the average Poisson rate. Although this might be of little concern for many applications in personal lines insurance, it is something we need to keep in mind for applications where data are sparse, such as commercial insurance and reinsurance.

2.1.8 Comparison of different techniques for supervised learning

We now collect our thoughts and considerations on the different techniques for supervised learning that we have investigated so far, in order to give an at-a-glance comparison among these techniques.

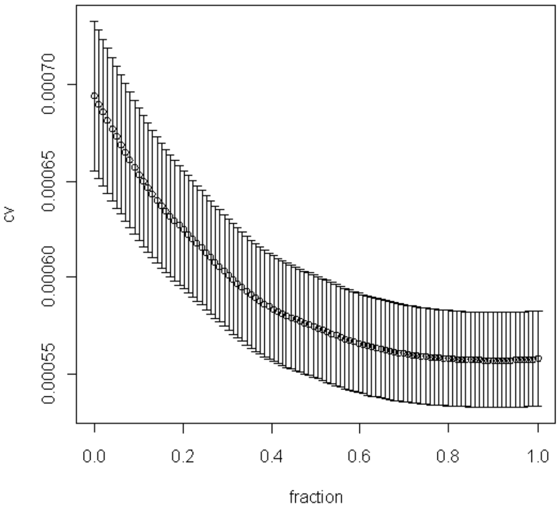
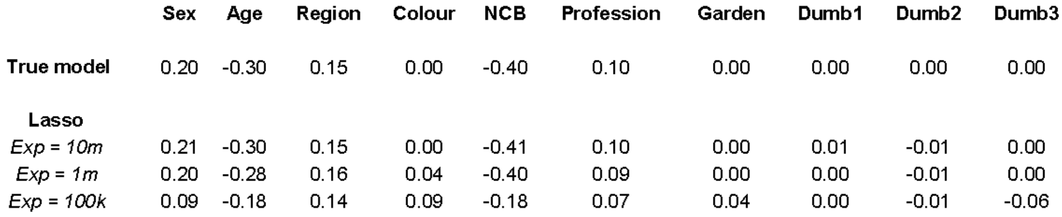


Figure 2.9: Expected prediction error as a function of the regularisation parameter for the lasso, in the case the exposure is $E = 100,000$ (average Poisson rate is around 3 for each category). Notice the unusual flattening when the value of the parameter reaches 0.8.



	Sex	Age	Region	Colour	NCB	Profession	Garden	Dumb1	Dumb2	Dumb3
True model	0.20	-0.30	0.15	0.00	-0.40	0.10	0.00	0.00	0.00	0.00
Lasso										
Exp = 10m	0.21	-0.30	0.15	0.00	-0.41	0.10	0.00	0.01	-0.01	0.00
Exp = 1m	0.20	-0.28	0.16	0.04	-0.40	0.09	0.00	0.00	-0.01	0.00
Exp = 100k	0.09	-0.18	0.14	0.09	-0.18	0.07	0.04	0.00	-0.01	-0.06

Table 2.4: Coefficients of the best lasso model for different values of the exposure. Notice how the different between the true coefficients and the calculated ones become larger as the exposure decreases.

Neural networks

↑ Neural networks are flexible: they are not limited by linearity and they can model any function of the data

↑ They do not require detailed model specification and are therefore easy to implement

↓ They provide prediction without interpretation, and they are therefore hard to accept by underwriters, management, regulators. They do not help us understanding risk.

↓ They are computationally demanding.

Generalised linear models

↓ Generalised linear models are limited by linearity. However, in practice, a large dictionary of functions is possible and linearity in the coefficient should not usually be crippling

↑ The loss function is flexible and can be chosen to reflect the underlying process noise

↑ Their interpretation is easy

↓ Model selection with forward or backward selection is computationally awkward and may get trapped in local minima

↓ Standard methods for model validation (as those based on AIC) may overestimate the test error

Regularised regression

↑ Regularised regression can be used to address sparsity (lasso regression, elastic net)

↑ Model selection and validation can be performed very efficiently, avoiding the combinatorial explosion of best subset selection

↓ The use of quadratic loss function for classical regularised regression is limited and may cause degradation of performance, for example for Poisson noise with low Poisson rates. This can be amended by using a different loss function, but some of the theoretical results obtained for squared loss (e.g. consistency) may be lost, and the solution may not be as simple[2].

↑ The interpretation of regularisation results is easy as for GLM

↑ Specific types of regularisation such as the elastic net can address the problems arising when there are highly correlated variables and when the number of predictors far exceeds the number of available data points

The comments in this section reflect the “textbook approach” of the relevant methodologies. It is fair to mention, however, that these methodologies come in many flavours and that hybrid approaches are possible and have actually been attempted.

As an example, GLM can be coupled with regularisation: see for example [PH96], where L_1 regularisation (lasso) was applied to a $\log P$ loss function. In my view this is one of the most promising avenues that actuaries should explore.

Also, cross-validation is used more in the context of neural networks and regularised regression than in the context of GLMs, but there is no reason why it should not be used for GLM as well.

The main message is that actuaries should keep on open mind on all these techniques and combine them freely as needed.

2.2 Unsupervised learning

Informally (and loosely), unsupervised learning is finding patterns in data without a ”teacher”, i.e. someone who can confirm whether the patterns we have found correspond to something of interest – as opposed to being random fluctuations. To make a simple example, suppose you are interested in grouping people in a certain country (let’s say Italy) according to the dialect they speak, based on a number of unlabelled recordings of their speech

[2] The role of the loss function in regularisation is investigated further in [RDVC$^+$04], to which the interested reader is referred.

that have been delivered to you by mail. You may select a number of features to analyse, perhaps properties of the frequency spectrum.

If you have an interpreter, you can classify each recording by giving it the proper label: e.g.. Piedmontese, Ligurian, Lombard, etc. When you are presented with a new speech sequence, you'll then be able to attempt classifying as one of the previously labelled dialects based on an algorithm developed during the training stage. This is supervised learning.

In unsupervised learning, you may want to solve the same problem, but this time there's no one who can help you labelling each speech sequence (this example is of course quite artificial as one would be led at least by geography!). However, you'll still be able to perceive differences between the speeches and your algorithm is likely to detect similarities between people using the same dialect. This is unsupervised learning. Actually there is a bit of cheating here because the underlying categories (the dialects) are ultimately verifiable, whereas in reality we are seldom that lucky – and as soon as get lucky, it's supervised learning again.

Formally, unsupervised learning can be defined as the following goal:

> Goal: given a set of observations $(X_1, \dots X_N)$ of a random p-vector X having joint density $Pr(X)$, infer the properties of $Pr(X)$ without the help of a teacher.

In unsupervised learning the target is unknown, and there are no dependent variables. This is to be compared to supervised learning, where one is concerned with determining $Pr(Y|X)$ from a training sample $(x_1, y_1), \dots (x_N, y_N)$, where the Y is the dependent variable.

The main problems of unsupervised learning are:

Clustering Find convex regions of the X-space containing modes of $Pr(X)$ (that is, can $Pr(X)$ be represented as a mixture of simpler densities?). More informally, this means segmenting data points into sets so that points in the same set are as similar as possible to each other and points in different sets are as dissimilar as possible.

Association rules Construct simple descriptions that describe regions of high density for high-dimensional (usually binary-valued) data (for example, market basket analysis)

Applications of unsupervised learning to general insurance problems are for example:

- Exploratory analysis, to check if there are patterns in data that may lead to further investigation.

- Descriptive data mining, which is actually a more organised version of exploratory analysis which is usually applied to large bases of data that would be otherwise difficult to make sense of.

- Clustering of policyholders, which attempts to label policyholders with similar behaviour with the idea of segmenting the account into different portions which will then be analysed later for, e.g., rating purposes..

In the following we will focus on clustering rather than on association rules: suffice it to mention that association rules can be useful, along with clustering, for data mining purposes.

2.2.1 Practical general insurance example: Territories clustering (Yao, 2008)

A simple, easy-to-visualise example is territories clustering as performed, for example, by Yao [Yao08]. In this case the input is – perhaps unusually – the output of a GLM exercise, where a number of features – excluding location – was used to produce a predictive model of claims experience for various categories of drivers. The reason why the author performs clustering is to have an extra variable describing the residual effect of location, and instead of subdividing regions into pre-defined regions (e.g. postcode group), or by type of location (city v town v rural...), or both, he creates his own territorial units by clustering.

Once each point on the map is assigned a label identifying the cluster, this label can be used as a rating factor, and a revised GLM exercise can be run with this extra-factor. This can be iterated until there is no further gain (hopefully, the expected prediction error will also decrease). We will not delve here into the relative merits of this methodology and a more traditional methodology which simply uses fixed locations as a rating factor – rather, we are interested in illustrating the results of this procedure.

Figure 2.10 shows the residuals of the GLM exercise. Notice the areas with a high density of black points (claims experience worse than predicted by GLM), such as London. What we want however is not clusters of points that have similar residual experience but clusters of points that are also close geographically. To obtain this effect Yao introduces a similarity measure between points which is a combination $f(x_1, x_2) + w \cdot g(x_1, x_2)$ of a geographical distance (f) and of a "distance" in claims experience (g).

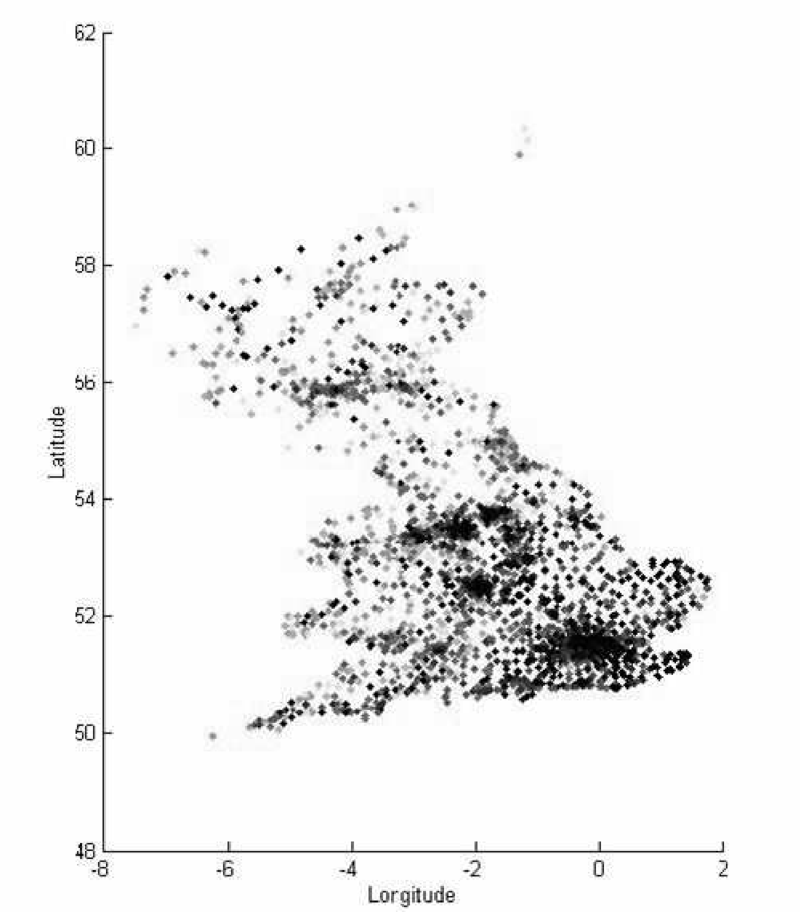


Figure 2.10: GLM residuals for each district (aggregation of postcodes). Black and blue indicate a claims experience worse than predicted by GLM. Green and yellow indicate a claims experience better than predicted by GLM. Other colours indicate a claims experience roughly similar to that predicted by GLM.

After the application of a well-known clustering technique (K-means – see Section 2.2.2), the clusters obtained in [Yao08] are those in Figure 2.11. Notice that despite the fact that some of these clusters have the same colour, all spatially separated should be considered different.

There is a large number of different clustering techniques. In the next section we are going to give a short account of some of the most used techniques.

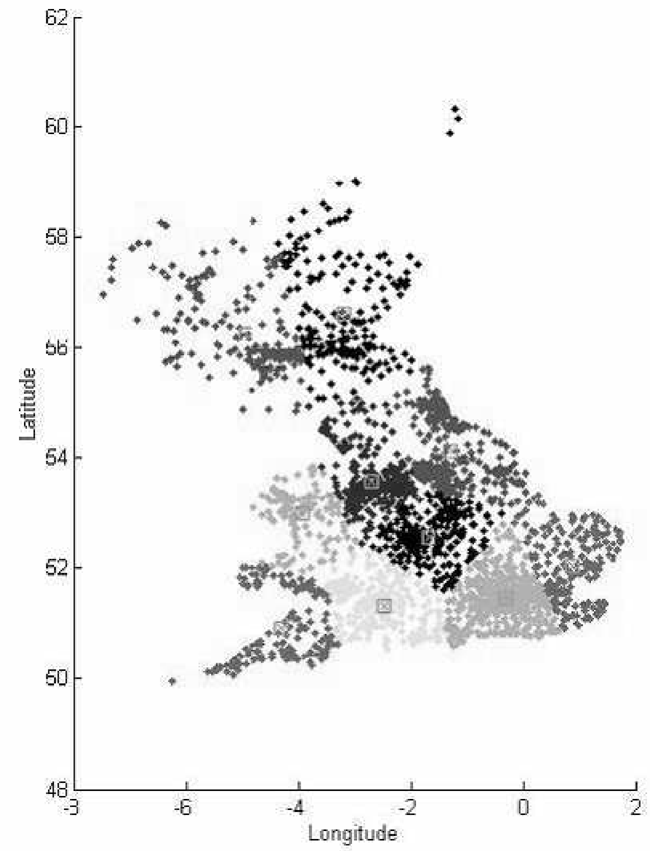


Figure 2.11: Clusters corresponding to regions of points which are close geographically and have homogeneous behaviour.

2.2.2 Clustering techniques

One possible categorisation of clustering techniques is this:

- Partitioning techniques (K-means, K-medoids, EM), which partition data based on a given dissimilarity measure;
- Hierarchical methods, which subdivide data by a successive number of splits;
- Spectral clustering, which works by graph partitioning techniques after a transformation into a suitable space;

- Other: density-based methods, grid methods, kernel clustering. These will not be dealt with here.

We are going to look at each of them in turn.

Partitioning techniques

Partitioning techniques subdivide data based on a given (dis)similarity measure. The most popular algorithm for clustering is perhaps K-means, as it is simple and intuitive.

The dissimilarity measure for K-means is simply the *squared* Euclidean distance $d(x_i, x_j) = ||x_j - x_i||^2$ (notice that x_i, x_j are in general vectors). The aim is to minimise the within-cluster point scatter:

$$W(C) = \frac{1}{2}\sum_{k=1}^{K}\sum_{C(i)=k}\sum_{C(j)=k} d(x_i, x_j) = \tag{2.29}$$

$$= \frac{1}{2}\sum_{k=1}^{K} N_k \sum_{C(j)=k} d(x_j, m_k) \tag{2.30}$$

The method works iteratively by

1. Choosing the number k of clusters
2. Choosing randomly (or cleverly) k data points as the initial cluster means
3. Assigning each object x to the closest cluster mean
4. Updating cluster means by minimising $W(C)$ with respect to the m_k
5. Iterating steps 3 and 4 until assignment does not change

Some advantages of this method are that:

- it is easy to understand and apply;
- converges always and quickly.

And some of the disadvantages are that:

- the number of clusters must be decided in advance. Of course, different values of K can be tried to check which value gives the “best” results;
- it is sensitive to noise, as it uses the squared distance between points, which can easily become large when a few points are added;
- it may get stuck in a suboptimal local minimum. As regards to this, it should be noted that the choice of the initial clusters is crucial;
- clusters lie in disjoint convex sets of the underlying space. This may be a disadvantage when the actual shape of the “real” clusters is unusual, as in Figure 2.13;
- there are issues when the density of different clusters is not uniform.

Notice that the clustering into territories shown in Figure 2.11 was obtained in [Yao08] by using k-means with the following dissimilarity measure:

$$g((x_i, y_i), (x_j, y_j)) + w \cdot f(m_1, E_1; m_2, E_2), \tag{2.31}$$

where $g((x_i, y_i), (x_j, y_j))$ is the Euclidean or the Haversine (more accurate geographically) distance; $f(m_1, E_1; m_2, E_2) = (m_1 - m_2)^2/(1/E_1 + 1/E_2)$, m_1, m_2 being the claim frequencies and E_1 and E_2 being the exposures. Figure 2.11 corresponds to the choice $w = 1$.

Examples of other partitioning methods (besides K-means) are K-medoids and expectation maximisation. K-medoids is very similar to K-means, but rather than choosing freely among points in the underlying space to be the cluster means, it always chooses one of the actual points in the dataset.

Expectation maximisation is a ubiquitous technique in statistics, that can be applied to a number of contexts, including clustering. See for example [HTF01] for an explanation.

Hierarchical methods

Hierarchical methods use cluster-to-cluster similarity in order to decide whether to merge/split clusters. They create a hierarchical decomposition forming a dendrogram (a tree that splits recursively). The idea behind a dendrogram is that depending on the dissimilarity threshold at which you decide that two clusters should be merged (y-axis), the number of clusters (i.e., the number of vertical lines crossed by a line with constant y) varies.

A dendrogram for the example of Figure 2.11 is shown in Figure 2.12.

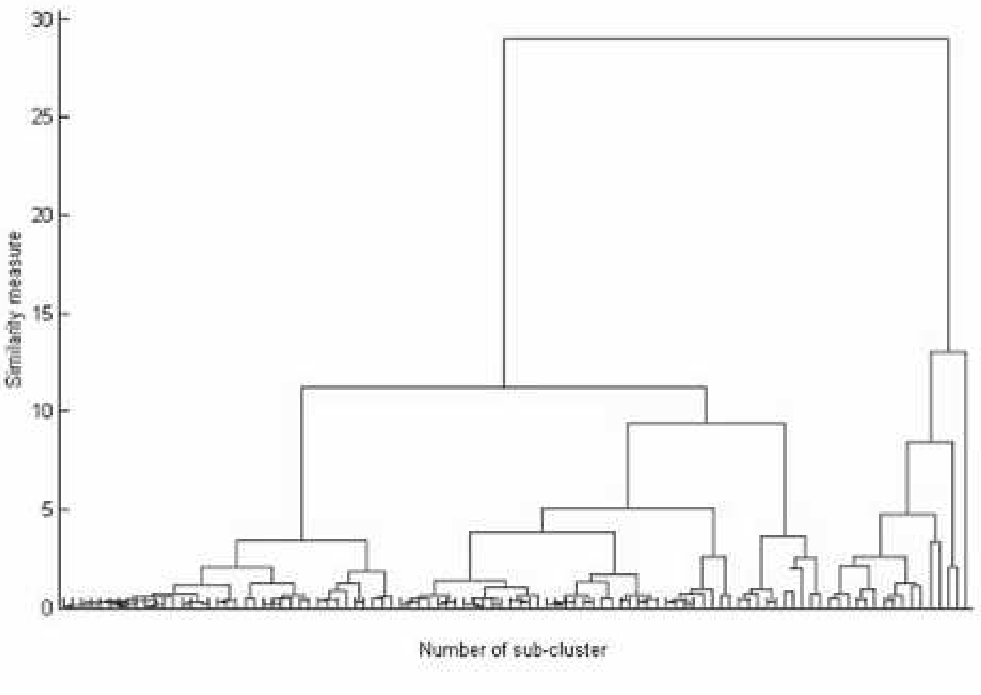


Figure 2.12: A dendrogram for territories clustering. Notice how the number of clusters depends on the dissimilarity measure at which two dendrograms are split.

Here are some characteristics of hierarchical methods:

- unlike k-means, they do not commit to a number of clusters K in advance. This gives the method more flexibility;

- they are easy to understand and apply;
- segmentation is irreversible – in the sense that once two clusters are merged, they will of course remain merged when we increase the threshold of the dissimilarity measure. In other terms, the merging is completely bottom-up;
- they are computationally demanding (in terms of time complexity).

Spectral clustering

Spectral clustering is becoming one of the most popular techniques in clustering, owing to its flexibility, its efficiency and its theoretical elegance. Only a cursory overview will be given here, but the interested reader will find an entertaining reference in the tutorial by von Luxburg [vL07].

Spectral clustering has its roots in graph theory, and its central idea is to solve the problem of clustering not in the original space (as K-means does) but on a transformed space embedded in a graph, the so-called similarity graph. The **similarity graph** is a graph whose vertices are the data points and two data points are connected by an edge if they are similar, as in Figure 2.13. The similarity function is somehow related to the "distance" between the points in a suitable space. For example, in a Euclidean space $\Re^k$ the similarity function might be defined as $s(x_i, x_j) = \exp(-\left\|x_i - x_j\right\|^2 / (2\sigma^2))$. Finding the clusters in the data set is then equivalent to finding partitions (groups of vertices) in this graph which are weakly connected to each other but whose vertices are very similar inside.

The essential aspect of spectral clustering is how you build the similarity graph, so it perhaps worth spending a few words on this. Three types of similarity graphs are commonly used:

- the **ϵ-neighbourhood graph**, in which every pair of vertices for which the similarity function is larger than ϵ is connected (see Figure 2.13(left)). This graph is usually unweighted;
- the **k-nearest neighbour graph**, in which the k nearest neighbours of a given vertex are connected to that vertex (see Figure 2.13(centre and right). Notice that v_j may be among the k nearest neighbours of v_i but not vice versa, therefore we have to decide whether to include the edge in this case: in the vanilla version of the kNN graph we include it, wehereas in the *mutual* version of the kNN graph we do not. This type

of graph is usually weighted, and the weight is the similarity between the vertices. This type of similarity graph is useful especially when we have clusters at different scales – that is, one cluster may be denser, the other more rarified, as in Figure 2.13;

- the **fully connected graph**, in which *all* vertices are connected. This graph will obviously be weighted and the weighting should decay exponentially, thus effectively defining a neighbourhood radius.

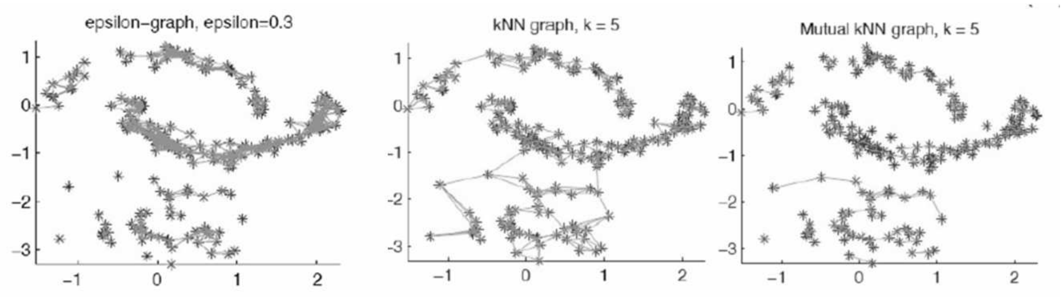


Figure 2.13: Different types of similarity graph: ϵ-neighbourhood graph (left), k-NN graph (centre), mutual k-NN graph (right). Notice the non-convex shape of the clusters and the fact that the clusters have different densities. The figure is borrowed from [vL07].

One version of the spectral clustering algorithm ("unnormalised spectral clustering") works as follows:

1. Build the similarity graph $G = (V, E)$ where $V = \{v_1, \ldots v_n\}$ are the vertices (the data points), E is a set of edges, that is a subset of $V \times V$ (pairs of vertices). The graph is assumed to be undirected. If the graph is weighted, that means that every edge (v_i, v_j) will have an associated weight $w_{i,j}$. The matrix $W = \{w_{i,j}\}_{i,j=1,\ldots n}$ is called the weighted adjacency matrix.

2. Calculate the graph Laplacian $L = DW$, where W is the adjacency matrix defined above and

$$D = diag(d_1, \ldots d_n), \text{where } d_i = \sum_j w_{i,j}$$

The laplacian L has many interesting properties: for example, it is symmetric and positive semi-definite, and has n non-negative, real-valued eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. The multiplicity k of the eigenvalue 0 of L equals the number of connected components $A_1, \dots A_k$ in the graph.

3. Calculate the first k eigenvectors $u_1, \dots u_k$ of L and build the matrix $U = (u_1, \dots u_k)$ (columns) $= (y_1, \dots y_n)^T$ (rows)

4. Group points $y_1, \dots y_n$ into k clusters $C_1, \dots C_k$ using, e.g., K-means

This algorithm produces sensible clusters, but it is not obvious why! Several arguments have been put forward to explain why spectral clustering works: one based on graph cuts, another on random walks, and yet another on perturbation theory. The perturbation theory point of view is probably the most intuitive and we outline it here. The tutorial by von Luxburg [vL07] explains all three points of view more at length.

Perturbation theory studies how the behaviour of eigenvalues and eigenvectors changes if one applies a small perturbating matrix H to the original matrix A. This may help us understand the workings of spectral clustering by considering the ideal case – that is, the case where the clusters are already perfectly separated at the stage when one builds the similarity graph. In this case, the points $y_1, \dots y_n$ built by the algorithm above will have the simple form $(0, \dots 0, 1, 0, \dots 0)$, and the position of the 1 indicates the connected component this point belongs to. All the vectors y_i belonging to the same connected component are coincident, and identifying the clusters is trivial.

If we now perturb the original matrix by a small amount, the points y_i will not be exactly $(0, \dots 0, 1, 0, \dots 0)$ as above, but a K-means algorithm will still find the same clusters.

Pros and cons of spectral clustering

The pros and cons of spectral clustering as listed below are mostly borrowed from [vL07].

- No strong assumptions on the shape of the clusters are needed. In particular, it is not necessary for clusters to be convex in the original space (see for example the moon-slice-shaped clusters in Figure 2.13).

- Spectral clustering is efficient on sparse graphs – of course, it is up to us to ensure that we choose the similarity graph so that it *is* sparse!

- Finding the solution given the similarity graph is a linear problem and therefore one cannot get stuck in local minima. Also, it is not necessary to restart the algorithm several times with different initial values, as for many other clustering methods.

- On the downside, the choice of a good similarity graph is crucial and not always easy. It is unstable under different choices of the parameters for the neighbourhood graph. All this amounts to saying that spectral clustering cannot be used as a black box!

- Spectral clustering may be difficult to understand/communicate: the very reason why it works is not obvious. However, its obscurity is not of the same type as that of neural networks, because here it is clear what the algorithm is trying to achieve – finding clusters on the similarity graph rather than on the original space.

2.2.3 Practical general insurance example: Descriptive data mining (Guo, 2003)

Another example of unsupervised learning which may use both clustering and association rules is data mining, which as we mentioned is the efficient discovery of previously unknown patterns in large databases. Data mining is used by many large companies (not necessarily, and not especially, in the financial sector) to make sense of their large collection of data and spot business opportunities.

Specifically, descriptive data mining is concerned with segmentation, link analysis, deviation detection, etc. It is especially useful in high-dimensional spaces, where it would be difficult to find appropriate statistics. It is based on clustering techniques and association rules.

Data mining has been studied in the actuarial literature, for example in [Guo03]. It is a vast subject and here we just wish to outline the data mining process works and give a couple of examples of applications.

The process works iteratively as follows:

1. Data acquisition (select the types of data to be used)

2. Pre-processing (data cleaning, transforming, etc)

3. Data exploration and modeling building

   - Choose data mining operation (e.g. classification, regression, segmentation, link analysis, deviation detection)
   - Select techniques (e.g. Bayesian analysis, artificial neural networks, genetic algorithms, decision trees) and algorithms
   - Perform data mining

4. Interpretation/evaluation
   - Filter redundant/irrelevant patterns
   - Visualise graphically/
   - Determine/resolve conflicts

Here are some example of successful applications of data mining to general insurance:

- descriptive data mining has allowed to discover surprising relationships, e.g. the fact that the credit rating of an individual is strongly linked to the individuals propensity to claim;
- for one insurer it has been found out by this method that a subset of a very risky category (males less than 21 y.o.) in its portfolio, those that have vintage cars, are far less risky than average (Figure 2.14).

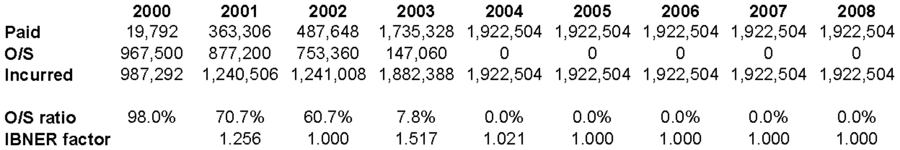
2.2.4 Practical general insurance example: Clustering for IBNER data

A third example of the possible use of unsupervised learning is in IBNER calculation. The underlying problem is that of predicting the ultimate value of a claim (or a probability distribution) given its previous history of paid and incurred amounts. An example is shown in Figure 2.15.

The numerical example illustrated here was analysed with the help of Nicola Rebagliati, from the Department of Computer Science of the University of Genoa.

For pricing or reserving purposes, especially for large losses, one often looks at the distribution of past IBNER factors to predict the ultimate value of a particular claim or to put aside some IBNER reserves. The idea is that by

Figure 2.14: A good risk.



	2000	2001	2002	2003	2004	2005	2006	2007	2008
Paid	19,792	363,306	487,648	1,735,328	1,922,504	1,922,504	1,922,504	1,922,504	1,922,504
O/S	967,500	877,200	753,360	147,060	0	0	0	0	0
Incurred	987,292	1,240,506	1,241,008	1,882,388	1,922,504	1,922,504	1,922,504	1,922,504	1,922,504
O/S ratio	98.0%	70.7%	60.7%	7.8%	0.0%	0.0%	0.0%	0.0%	0.0%
IBNER factor		1.256	1.000	1.517	1.021	1.000	1.000	1.000	1.000

Figure 2.15: The history of a claim occurred in 2000 and reported in the same year. For each year of development, the paid, outstanding and incurred (estimated) amounts are given. The outstanding ratio (outstanding amount divided by incurred amount) is also given for each year. The last row shows the IBNER factors for this claim, calculated as the ratio between the incurred amounts of two successive years. The claim shown here is a real claim after scaling by a random amount and the removal of any identifiable information.

spotting a systematic bias (on average) on the claims estimates, the estimates can be put right (again, on average!) by correcting all claims with the same characteristics.

The calculation of IBNER factors is usually performed by chain ladder calculations (or something similar) and often a simple average over all open claims (settled claims will of course have no IBNER, unless they are re-opened) is all that is used, especially if the number of individual claims histories is not large: the output of such analysis would then be an IBNER factor which depends only on the development year.

However, it is obvious that the IBNER factor will depend in general on many factors, such as:

- development year (the younger the claim, the more scope there is for it to deviate from the incurred estimate), as already mentioned;
- size of claim (larger claims may have more uncertain about them)
- outstanding ratio (the larger the outstanding amount, the more conjectural the estimate of the incurred value will be);
- type of claim (for example, bodily injury claims will be more difficult than property claims to estimate reliably as the victim's health may worsen unexpectedly);
- reserving style (claims managers may be more or less effective at reserving, and some may introduce a systematic upward/downward bias);
- ...

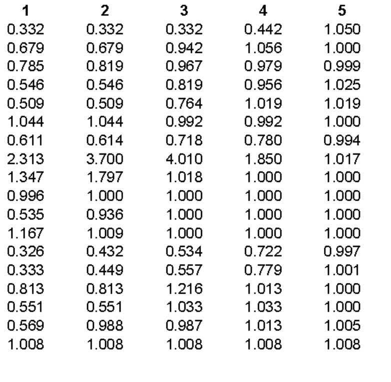
As all situations in which one needs to understand how different factors impact a given variable, one can use the data set to perform supervised learning as in Section 2.1. However, not all the variables in the list above lend themselves to be used easily as input variables: specifically, the "reserving style" above is quite difficult to use as the claims history is a vector of, say, 15 years of development.

One possible solution is to collect all individual claims development histories and group them into clusters of similar behaviour. Once clustering is performed, the cluster ID can be used as one of the rating factors in a regression exercise with GLM or regularisation.

One complication of this exercise is that different claims histories will have different lengths (some may be 2 years long, some 15) and therefore one must put a little care into how one defines a similarity measure. Since here we are only interested in giving an illustration of various applications of clustering, we will limit ourselves to extract sections of claims histories of exactly 5 years (the history of the claim may go on after 5 years, but we look at a section of the data spanning 5 years only). Due to the nature of the losses (large losses that appear in the database only once they become greater than a given reporting threshold) the development year is calculated from the year of exceedance rather than from the year of occurrence.

The data used are large motor losses from several UK companies. The losses have been normalised so that the latest known incurred value (not necessarily that at the fifth year) is set to 1 – which will make them unrecognisable as well as removing the dependency on loss size.

Figure 2.16 shows a few examples of the claims histories we are trying to cluster, after normalisation.



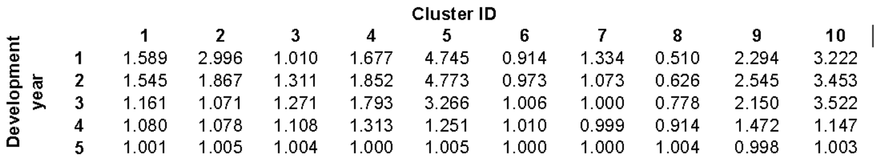
1	2	3	4	5
0.332	0.332	0.332	0.442	1.050
0.679	0.679	0.942	1.056	1.000
0.785	0.819	0.967	0.979	0.999
0.546	0.546	0.819	0.956	1.025
0.509	0.509	0.764	1.019	1.019
1.044	1.044	0.992	0.992	1.000
0.611	0.614	0.718	0.780	0.994
2.313	3.700	4.010	1.850	1.017
1.347	1.797	1.018	1.000	1.000
0.996	1.000	1.000	1.000	1.000
0.535	0.936	1.000	1.000	1.000
1.167	1.009	1.000	1.000	1.000
0.326	0.432	0.534	0.722	0.997
0.333	0.449	0.557	0.779	1.001
0.813	0.813	1.216	1.013	1.000
0.551	0.551	1.033	1.033	1.000
0.569	0.988	0.987	1.013	1.005
1.008	1.008	1.008	1.008	1.008

Figure 2.16: Examples of loss progressions. The most recent estimate of the claim (possibly beyond the 5-year horizon) is normalised to 1.

The best results were obtained with $K = 10$ clusters. The cluster means are shown in Figure 2.17.

Notice that these clusters can be interpreted as follows:

- cluster 6 is the most typical behaviour and occurs when the estimate of the incurred amount is quite stable after an initial 8% increase. Cluster 7 is also relatively stable but it starts with a 30% over-reserving;

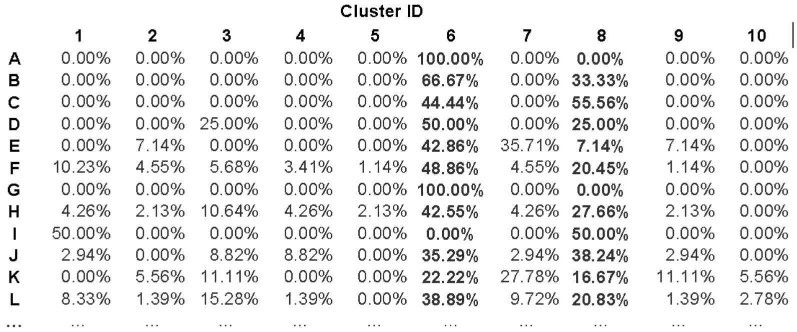


		Cluster ID									
		1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Development year	**1**	1.589	2.996	1.010	1.677	4.745	0.914	1.334	0.510	2.294	3.222
	2	1.545	1.867	1.311	1.852	4.773	0.973	1.073	0.626	2.545	3.453
	3	1.161	1.071	1.271	1.793	3.266	1.006	1.000	0.778	2.150	3.522
	4	1.080	1.078	1.108	1.313	1.251	1.010	0.999	0.914	1.472	1.147
	5	1.001	1.005	1.004	1.000	1.005	1.000	1.000	1.004	0.998	1.003

Figure 2.17: Cluster centres for the clusters calculated with K-means, $k = 10$.

- cluster 8 shows a typical pattern of under-reserving;
- clusters 1, 2, 4, 5, 9, 10 are typical patterns of over-reserving;
- cluster 3 shows claims for which the incurred amount has returned to the initial value after a period of over-reserving.

It is interesting to compare these clusters with a piece of information that we have not used in the clustering process – the name of the company that reserved those claims. The results of this comparison are shown in Figure 2.18. It is clear that certain companies (such as Company J) have a higher-than-average tendency to under-reserve, and this is confirmed by what one knows about these companies.



	Cluster ID									
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
A	0.00%	0.00%	0.00%	0.00%	0.00%	**100.00%**	0.00%	**0.00%**	0.00%	0.00%
B	0.00%	0.00%	0.00%	0.00%	0.00%	**66.67%**	0.00%	**33.33%**	0.00%	0.00%
C	0.00%	0.00%	0.00%	0.00%	0.00%	**44.44%**	0.00%	**55.56%**	0.00%	0.00%
D	0.00%	0.00%	25.00%	0.00%	0.00%	**50.00%**	0.00%	**25.00%**	0.00%	0.00%
E	0.00%	7.14%	0.00%	0.00%	0.00%	**42.86%**	35.71%	**7.14%**	7.14%	0.00%
F	10.23%	4.55%	5.68%	3.41%	1.14%	**48.86%**	4.55%	**20.45%**	1.14%	0.00%
G	0.00%	0.00%	0.00%	0.00%	0.00%	**100.00%**	0.00%	**0.00%**	0.00%	0.00%
H	4.26%	2.13%	10.64%	4.26%	2.13%	**42.55%**	4.26%	**27.66%**	2.13%	0.00%
I	50.00%	0.00%	0.00%	0.00%	0.00%	**0.00%**	0.00%	**50.00%**	0.00%	0.00%
J	2.94%	0.00%	8.82%	8.82%	0.00%	**35.29%**	2.94%	**38.24%**	2.94%	0.00%
K	0.00%	5.56%	11.11%	0.00%	0.00%	**22.22%**	27.78%	**16.67%**	11.11%	5.56%
L	8.33%	1.39%	15.28%	1.39%	0.00%	**38.89%**	9.72%	**20.83%**	1.39%	2.78%
...	...	...	...	...	...	...	...	...	...	...

Figure 2.18: The percentage of each type of cluster for a selection of UK companies.

It is interesting to compare these results with those one obtains with spectral clustering. To use spectral clustering, we first have to choose a similarity graph. One choice is the ϵ-neighbourhood graph with similarity function $s(x_i, x_j) = \exp(-\|x_i - x_j\|^2/(2\sigma^2))$, $\|x_i - x_j\|$ being the distance in $\Re^5$ between our vectors representing the development of claims. However, this is going to give results very similar to the K-means algorithm tested above.

We have therefore used the k-nearest-neighbour similarity graph. The choice of k is of course crucial. Unfortunately, there are few theoretical results guiding us in the choice of k. A few heuristics are often recommended, though. One is that k be chosen so that the resulting similarity graph is connected, or that the number of connected components is less than the number of expected clusters – otherwise there is a danger that the algorithm will simply return the connected components as clusters. For very large graphs, a rule of thumb is to look at values of k in the region of $\log n$ (n being the number of data points). In our case, this does not give sensible results.

Another option is to look at the so-called Laplacian spectrum, which gives (for a given number of k of the kNN graph) the value of all the n eigenvalues (some of them possibly repeated if their eigenspace has dimension > 1) in decreasing order. In the ideal situation we would have only K eigenvalues[3] that are near zero and can therefore be considered – from a perturbation theory point of view – as "noisy" versions of the first eigenvalue $\lambda_1 = 0$ of multiplicity K. Then there should be, always in the ideal case, a gap between the K-th and the $K+1$-th eigenvalue, related to the gap between λ_1 and λ_2.

After trying out several values of k, we have found that we need to go up to about $k = 300$ to have the appropriate structure of the graph, and the Laplacian spectrum indicates that a reasonable choice for the number of clusters is 5. See Figures 2.19 and 2.20 for two examples of Laplacian spectrum.

Notice that the selected number ($k = 300$) is quite an extreme choice for the number of nearest neighbours in the similarity graph. The underlying reason might just be that there are no natural clusters in this IBNER problem, because the patterns in IBNER development are not sharply defined. However, the existence of a few (3-5) patterns is indeed confirmed by a comparison of the results of spectral clustering and K-means, and by looking at which clusters are more common for each company.

These results are shown in Figure 2.21, which shows the centres of the five clusters found with spectral clustering, and in Figure 2.22, which shows the percentage of each cluster for each company. There is indeed a striking sim-

[3] K indicates here the number of clusters, k is the number of nearest neightbours in the kNN graph.

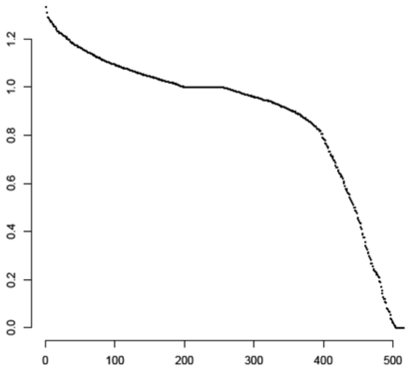


Figure 2.19: Laplacian spectrum for the k-nearest-neighbour similarity graph, $k = 20$. No structure is apparent in the graph and there is a large number of eigenvalues near zero, without a clear difference between the near-zero behaviour and the rest. It is therefore not possible to identify a small number of eigenvalues that are near-zero.

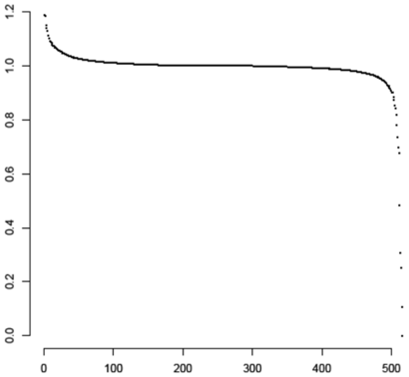
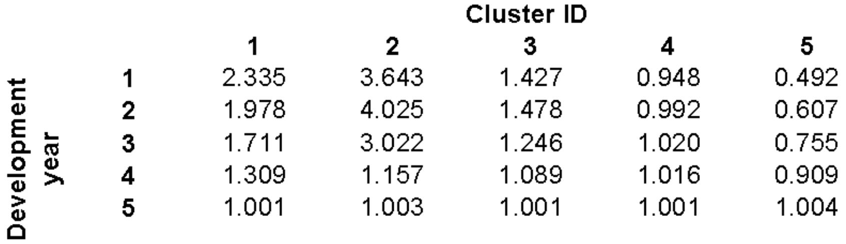


Figure 2.20: Laplacian spectrum for the k-nearest-neighbour similarity graph, $k = 300$. The structure is better and there are only a few eigenvalues below 0.5 (five of them). There is then a significant gap between the fifth and the sixth eigenvalue. This structure therefore suggests the existence of five clusters (but anything between 3 and 5 would be acceptable).

ilarity with the results obtained with K-means: in both cases, we have an important cluster collecting all the claim developments that are stable (cluster 6 for K-means and cluster 5 for spectral clustering). Also, in both cases we have a single cluster which attracts all claims that are under-reserved and sometimes keep growing year-on-year. The other clusters are in both cases mostly examples of over-reserving and could easily be further aggregated. The main difference is the presence in K-means of a cluster (cluster 3) that starts from 1, goes up to 1.3 and goes back to 1 again – however, this is likely to be an artifact, and spectral clustering is probably right in ignoring it.

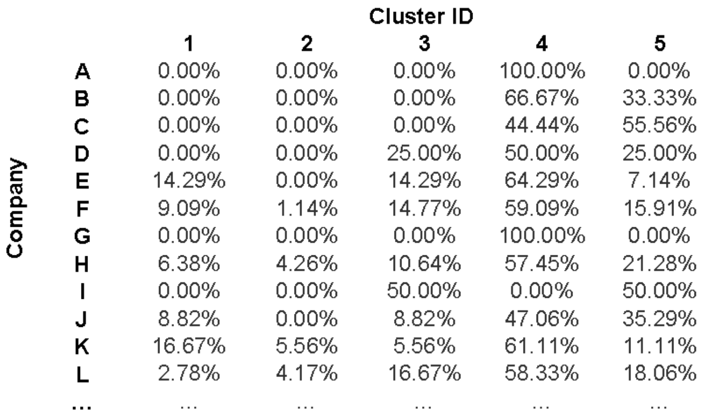
Overall, there seems to be some improvement with respect to K-means. However, this is probably not the happiest of examples because the clusters are probably already convex in the original space, and clusters are not sharply defined because claims developments vary over a continuous range.



				Cluster ID		
		1	2	3	4	5
Development year	1	2.335	3.643	1.427	0.948	0.492
	2	1.978	4.025	1.478	0.992	0.607
	3	1.711	3.022	1.246	1.020	0.755
	4	1.309	1.157	1.089	1.016	0.909
	5	1.001	1.003	1.001	1.001	1.004

Figure 2.21: Cluster centres for the clusters calculated with spectral clustering. A k-nearest-neighbour similarity graph has been used, with $k = 300$.

Although in this case the difference between K-means and spectral clustering is not striking, spectral clustering is a far more flexible method (which incorporates K-means anyway in the standard implementation of the method) and despite the fact that it takes a while to explain, it is quite simple to implement, it is very efficient and has some important properties (local minima, convexity in the appropriate space). It is therefore something that actuaries should consider alongside k-means when doing clustering, especially when the structure of their data is complex, before concluding that clustering does not help them.



				Cluster ID		
		1	2	3	4	5
Company	A	0.00%	0.00%	0.00%	100.00%	0.00%
	B	0.00%	0.00%	0.00%	66.67%	33.33%
	C	0.00%	0.00%	0.00%	44.44%	55.56%
	D	0.00%	0.00%	25.00%	50.00%	25.00%
	E	14.29%	0.00%	14.29%	64.29%	7.14%
	F	9.09%	1.14%	14.77%	59.09%	15.91%
	G	0.00%	0.00%	0.00%	100.00%	0.00%
	H	6.38%	4.26%	10.64%	57.45%	21.28%
	I	0.00%	0.00%	50.00%	0.00%	50.00%
	J	8.82%	0.00%	8.82%	47.06%	35.29%
	K	16.67%	5.56%	5.56%	61.11%	11.11%
	L	2.78%	4.17%	16.67%	58.33%	18.06%
	...	...	...	...	...	...

Figure 2.22: A comparison of a few companies (after scrambling, and selection of the first half of them, to make them fully unrecognisable).

Chapter 3

Dealing with uncertain and soft knowledge

> ”Do you want all the works?”
> ”On the record. Off the record. All the maybes.”
> *Smiley’s people, John Le Carré*

A significant portion of what we know about risk is

- Uncertain
- Qualitative (“soft”)
- Fuzzy
- Anecdotal

One does not need to look hard to find examples where uncertain and soft knowledge is significant in insurance. In one of the simplest, textbook problems of general insurance, the estimation of the prospective loss ratio based on historical loss ratios and other relevant information, one is faced with all the issues above at the same time (see Section 1).

Another classical example is the determination of the level of reserves: along with all the uncertainties of our projection techniques, both in the choice of the model and in the choice of the parameters, we have to take into account what we know about, say, impending legislation, past examples (e.g. asbestosis claims), etc.

We have several types of uncertainty that we need to address in actuarial practice:

- **Process uncertainty**: the uncertainty that derives from dealing with inherently stochastic phenomena, or phenomena that appear to be stochastic at the level we are able to look at them. Even if we knew for sure that the number of claims for an account can be modelled by a Poisson distribution and we knew the Poisson rate with infinite accuracy, the actual number of claims would have random fluctuation from one period to the other.
- **Parameter uncertainty**: the additional uncertainty that derives from the fact that the parameters of our models are never known with 100% accuracy, even if the model is correct. In the case mentioned above, we may know for sure that our claims come from a Poisson process, but we will know what the rate is only with limited accuracy.
- **Model uncertainty**: the additional uncertainty that comes from not being sure what the correct model is. In the example above, we may not be fully sure that the process is Poisson: it might be a negative binomial distribution instead.
- **Data uncertainty**: the additional uncertainty that comes from knowing the data only with limited accuracy. For example, in the process above not all claims may be reported immediately, and therefore in every time period the number of claims will only be an estimate, based among other things on our estimates of IBNR claims.

We know how to deal with process uncertainty and with parameter uncertainty with statistics. However, the treatment of model uncertainty and data uncertainty is less developed.

One may legitimately suspect that model uncertainty cannot really be quantified rigorously, because being able to quantify model uncertainty requires no less than knowing *what the correct model is.* However, we will see that some results are available, at least when the choice is between a finite number of models and it is possible to make some assumptions on how likely a given model is to be valid.

The case of data uncertainty is quite interesting because it comprises in a way all kind of soft knowledge that we know about a problem, and it seems that there are different approaches that are competing to be heard. Among these are fuzzy set theory, Bayesian analysis, rule-based systems, non-monotonic logic, belief theory. On some of these approaches, there is already a body of actuarial literature.

On the up side, a large lore is usually available to risk professionals. This lore – which includes everything from market information to "hunches" – is itself mired in uncertainty, but can also be used to reduce the range of possible estimations of risk.

Section 3.1 will deal with fuzzy set theory. The Bayesian approach will be discussed in Section 3.2, and a specific Bayesian approach (that of Bayesian networks) will be described in Section 3.3. A brief discussion of rule-based systems and non-monotonic logic will be attempted in Section 3.4

Section 3.5 provides a simple worked-out example that shows how to deal with data uncertainty and prior knowledge with fuzzy set theory, Bayesian analysis (and Bayesian networks), rule-based systems.

3.1 Fuzzy set theory approach

The motivation behind fuzzy set theory [Zadeh, 1965] is the attempt to capture and handle analytically the notion of an object (e.g. a number, or a logical value) whose value is not sharply defined: e.g. a bad winter, "a large loss", "a risky account".

At the core of fuzzy set theory, which is a branch of set theory, is the concept of **fuzzy membership**: a fuzzy set A in Ω is a set of ordered pairs $A = \{x, \mu_A(x)\}$, where $x \in \Omega$, and $\mu_A : X \to [0,1]$ gives the degree of membership of x to the set A. This is a simple and natural generalisation of the concept of belonging to a set: in the "vanilla" version of set theory, an element either belongs to a set or it doesn't: the value of the membership function is either 1 or 0. Here, a gray area is possible.

Fuzzy set operations can also be defined as a natural extension of the operations for standard set theory:

$$\begin{aligned} A \subseteq B &\iff \forall x, \mu_A(x) \leq \mu_B(x) \\ \mu_{\overline{A}}(x) &= 1 - \mu_A(x) \\ \mu_{A\cup B}(x) &= \max\{\mu_A(x), \mu_B(x)\} \\ \mu_{A\cap B}(x) &= \min\{\mu_A(x), \mu_B(x)\} \qquad (3.1) \end{aligned}$$

The definition of the intersection as given above actually can give rise to counterintuitive results in some application, and some authors (see for example [Lem90]) have tried to amend it to achieve some desired effect.

Fuzzy numbers can also be defined based on membership functions. In informal terms, a fuzzy number is a fuzzy subset of $\Re$ whose membership function is centered around a given real number. In other words, fuzzy numbers are a fancy version of the "ranges" much used by actuaries, the main difference being that they come with an exact rule that assigns more membership to some central value (ranges do the same – the central value being the best estimate – but in a more vague fashion).

A simple example of fuzzy numbers in general insurance is provided by the individual loss estimates for claims that have been reported but have not yet been settled. Ignoring loss adjustment expenses, this will usually be a fuzzy number with lower limit equal to the amount paid that far (although recoveries are possible, so it could be argued that the inferior limit is always zero), and the higher limit will be the policy limit (if this exists); or a more strict range if the claim manager is confident about the possible outcomes. The highest membership level is attained at the best estimate. This is an example of how fuzzy numbers can be used to represent uncertain knowledge.

Fuzzy numbers can also be used to represent soft or fuzzy knowledge. For example, when doing exposure rating for property business, the exposure curve must be chosen to reflect the type of business underwritten by the insurance company. In a frequently used setting, one will use Swiss Re curves with different values of c, where the parameter c is related to the concavity of the exposure curve. For high values of c, the concavity is very pronounced, implying that the event of a total loss is unlikely (as is usually the case for large industrial property risk), whereas a small value of c indicates a very high likelihood of a total loss, as is the case for small residential properties. Lacking adequate statistics on the property portfolio, the actuary is sometimes required to price property business based on the underwriter's perception of the "heaviness" of the portfolio. So the actuary might have to price a "fairly light portfolio, with mainly residential buildings and small shops". This knowledge may be incorporated by taking c to be a fuzzy number, with values in a reasonable range, say $c \in (1.5, 4.0)$.

Now that we have fuzzy numbers, we can create a **fuzzy arithmetic**. The essential tool for this is Zadeh's extension principle, by which if * is a binary operation (e.g. sum, product...) and A, B are two fuzzy numbers, then the membership function for $A \circledast B$ – the fuzzy number corresponding to the application of operation "*" – is

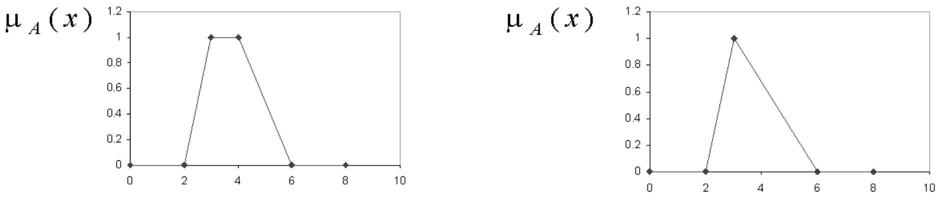


Figure 3.1: Two common examples of fuzzy numbers. Triangular fuzzy numbers (right) are a popular choice for modellers.

$$\mu_{A\circledast B}(z) = \sup_{x,y}\{\min[\mu_A(x), \mu_B(x)]\}, \text{ where} z = x * y \tag{3.2}$$

As an example, the sum $A \oplus B$ of two fuzzy numbers A, B is as in Figure 3.2. Much of the arithmetic of fuzzy numbers bears an obvious resemblance to interval arithmetic.

The definition for **crisp functions of fuzzy numbers** is similar to that of fuzzy operations:

$$\mu_{f(A)}(z) = \sup_{x\in\Re}\{\mu_A(x)|y = f(x)\} \tag{3.3}$$

Notice the different implications of the definitions above for a monotonic function f (for which there is only one value $x \in \Re$ such that $y = f(x)$, and therefore the prescription to take the supremum is superfluous), and for a non-monotonic function.

Notice that the sum of two triangular fuzzy numbers (a popular choice among modellers) is a triangular fuzzy number. This does not generalise to other elementary operations, such as the product of two fuzzy numbers.

Although the extension principle is simple enough when applied to monotonic functions and triangular fuzzy numbers, it becomes quickly unwieldy when applied to non-monotonic complex functions and to operations involving fuzzy numbers with complex membership functions. This is not only a theoretical point: the problem arises in all but the simplest toy problems. Whenever you apply a chain of operations on fuzzy numbers, e.g.

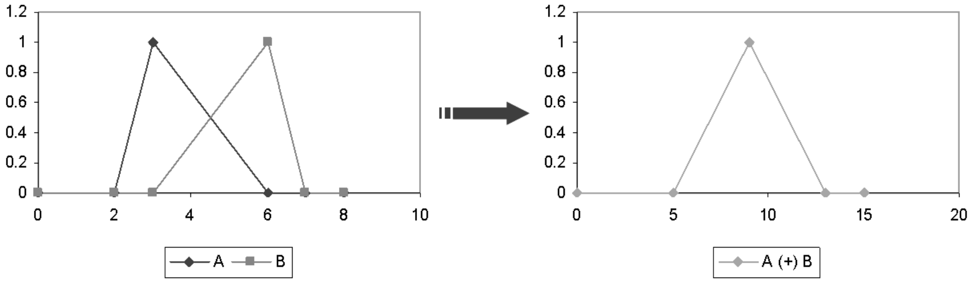


Figure 3.2: Sum of two fuzzy numbers A, B

$(f(A) \otimes B \oplus C) \otimes D$, it becomes quite difficult to determine what the final membership function is. It is perhaps for this reason that (as far as the author is aware of) a complete fuzzy calculus has never been developed.

3.1.1 Examples in the actuarial literature

Many attempts have been made by actuaries and researchers from other disciplines to apply fuzzy set theory to actuarial problems. In this section we very briefly sketch the content of a couple of papers written about these applications. The reader whose interest in the subject cannot be quenched by this very cursory treatment is referred to the paper by Shapiro [Sha04], which gives an overview of twenty years of fuzzy logic applications to insurance.

Of special interest is the 1997 paper by Cummins and Derrig [CD97], “Fuzzy financial pricing of property-liability insurance”, which includes an interesting discussion section with other authors on the merits of the fuzzy set theory approach. The paper attempts to “fuzzify” the Myers-Cohn method for pricing general insurance contracts, according to which the price of a policy is set so that “the net present value of premiums is equal to the present value of the losses, expenses and federal income taxes incurred by the company as a result of issuing a particular policy”. The general equation for the Myers-Cohn method is:

$$PV(P) = PV(L) + PV(\text{Tax}) \tag{3.4}$$

where $PV(\cdot)$ is the present value operator, P is the premium, L is the expected loss payment, and Tax is the federal tax on underwriting and investment income. To calculate the present value we need to calculate a risk-adjusted discount rate for losses. In [CD97], this is derived by a CAPM approach, by which:

$$r_L = r_f + \beta_L \cdot (r_m - r_f) \tag{3.5}$$

where r_m is the expected rate of return on the market portfolio, r_f is the default risk-free rate of interest, and $\beta_L = \mathrm{Cov}(r_L, r_m)/\mathrm{Var}(r_m)$. Notice that according to most authors, β_L is negative, therefore the losses will be discounted by less than the risk-free rate. Premiums, on the other hand, are usually discounted at the risk-free rate, on the assumptions that the premium flow is not risky.

For example, in the simple case where we have only one period in which premium is received at time 0 and losses are paid at time 1, Equation 3.4 becomes:

$$P = \frac{L}{1 + r_L} + \frac{\tau r_f S}{(1 - \tau)(1 + r_f)} \tag{3.6}$$

where S is the surplus (capital) committed by the insurer and τ is the corporate tax rate for both underwriting and investment profit. Notice that since the premium is assumed to be received at time 0, it is not discounted in Equation 3.6.

Uncertainties enters the picture:

- in the selection of a risk premium model: CAPM is one possible choice, but not analysts will agree that it is a good choice. This uncertainty has non-statistical elements;
- in the calculation of the loss discount rate, r_L. This uncertainty also has non-statistical elements;
- in the calculation of the losses L. This seems an inherently probabilistic uncertainty, but there are other elements to it: for example, we may have a limited trust in the loss database (data uncertainty);
- in the possible presence of judgmental prior information.

To take these uncertainties into account, the authors fuzzify premiums, losses, the risk-free rate, the risk-adjusted discount rate, while leaving the tax rate

τ and the surplus commitment S crisp and solve the fuzzy equivalent of Equation 3.4 to produce a fuzzy net present value. The final decision on the project (the project of selling a policy) is then based on a defuzzification procedure that we will not describe here.

Another interesting paper is that by Lemaire [Lem90], which describes the problem of finding the optimal XoL retention and has a fuzzy go at the definition of a preferred policyholder. The paper is interesting also because of the difficulties that the author obviously encounters in making the concepts of fuzzy set theory useful for his trade, and how he has to modify the definition of the intersection of two fuzzy sets to avoid counterintuitive results (see discussion above).

Fuzzy set theory can also be applied to decision theory in the form of fuzzy rules: see for example [Ben96].

3.2 The Bayesian approach

In the probabilistic approach, uncertain quantities are modelled by probability distributions. The least controversial example is perhaps parameter uncertainty: the mean of n independent variables drawn from the same distribution with finite variance σ^2 is a random variable with variance σ^2/n. Beliefs and prior knowledge are also modelled as probability distribution. As an example, future inflation of liability claims may be represented as a Gamma distribution centred around a given value (say 7%) with a given standard deviation (say 2%), reflecting expert opinion on the behaviour of the economy and on the judicial environment.

The logical leap between the frequentist approach to probability and the Bayesian approach of using probability to model uncertain knowledge is not conceptually obvious and the Bayesian approach is actually a long-disputed use of probability. We cannot enter in this philosophical debate, but there is a couple of points that are relevant to our discussions.

The first point relates to the interpretation of the Bayes theorem:

$$Pr(\theta|\underline{Z}) = \frac{Pr(\underline{Z}|\theta)Pr(\theta)}{Pr(\underline{Z})}, \tag{3.7}$$

which allows to calculate the posterior probability $Pr(\theta|\underline{Z})$ of θ (e.g., a parameter) conditional to $\underline{Z}$ (e.g., the data) as the probability $Pr(\underline{Z}|\theta)$ of $\underline{Z}$

conditional to θ times the prior probability $Pr(\theta)$ of θ. Formally, this is a straightforward consequence of the axioms of probability theory when using conditional probability. What is being disputed is the use of $Pr(\theta)$ to represent our initial beliefs on θ: that is, the use of probability theory to represent ignorance rather than an objective state of things: compare this to the frequentist interpretation of probability (probability as the limit of frequency for a large number of trials) and to Popper's propensity theory of probability [Pop59].

The second point is that in practice it may be difficult to have detailed prior knowledge on $Pr(\theta)$, and therefore all the derivations based on Bayes' theorem may be of little use.

Despite these issues the Bayesian approach has gained more and more acceptance in recent decades, undoubtedly aided by the availability of larger computing resources that have made it possible to calculate posterior probabilities in complex situations. Specifically, the use of this rule can easily be generalised to chains and to whole networks (Bayesian networks) which allow the propagation and the update of beliefs.

Actuaries are familiar with the Bayesian formalism. For example, credibility theory, a standard tool of general insurance, can be (and has been) framed in a Bayesian context: the information coming from the data is combined with that coming from an external source (which represents our prior knowledge), and the relative weight is determined by the balance of the variance of our data and the variance of the prior distribution.

3.3 Bayesian networks

In all but the simplest cases, there are many stages between the available data and the final decisions to make on pricing, reserving, capital to be held, etc. A priori knowledge and uncertainty is not limited to one stage but is used ubiquitously. For this reason it is important to be able to determine how uncertainty and beliefs are propagated across all stages of the methodology. Bayesian networks provide a way to do this. The most complete reference to Bayesian networks is probably [Pea88]. A good, shorter (and simpler) introduction can however be found in [RN03].

Bayesian networks provide a way of representing a joint probability distribution, highlighting the dependency relationships between the variables. Formally they are *directed, acyclic graphs*, i.e. collections of nodes and arcs where each arc has a direction and there are no cycles:

- Each node represents a random variable and has an associated quantitative probability information.
- Two nodes A and B are connected by an arc from A (the parent node) to B (the descendant) if B is dependent on A.

The cornerstone for the construction of a Bayesian network is the so-called *chain rule* which gives the joint probability distribution as a function of the conditional probabilities:

$$Pr(X_1, \dots X_n) = Pr(X_n | X_1, \dots X_{n-1}) Pr(X_{n-1} | X_1, \dots X_{n-2}) \cdots Pr(X_2 | X_1) Pr(X_1) \tag{3.8}$$

To specify a Bayesian network, we need to identify which dependencies in Equation 3.8 (which is fully general) can be dispensed with, in order to make the network as compact as possible. Therefore we want to replace Equation 3.8 with the sparser

$$Pr(X_1, \dots X_n) = \prod_{i=1}^{n} Pr(X_i | \text{parents}(X_i)) \tag{3.9}$$

In Equation 3.9, parents(X_i) is the set of variables on which X_i actually depends directly. This suggests another view of Bayesian networks (alternative to that of a joint probability distribution) as **a graph encoding a collection of conditional independence statements**. This encoding is called a "topological semantic" and can be expressed by one of the following equivalent specifications:

- a node is conditionally independent of its non-descendants given its parents;
- a node is conditionally independent of all the other nodes in the network, given its Markov blanket (= parents, children, children's parents)

An example of a Bayesian network is provided in Figure 3.3. This (simplified!) Bayesian network relates the total compensation for a bodily injury claim to the main characteristics of a claim.

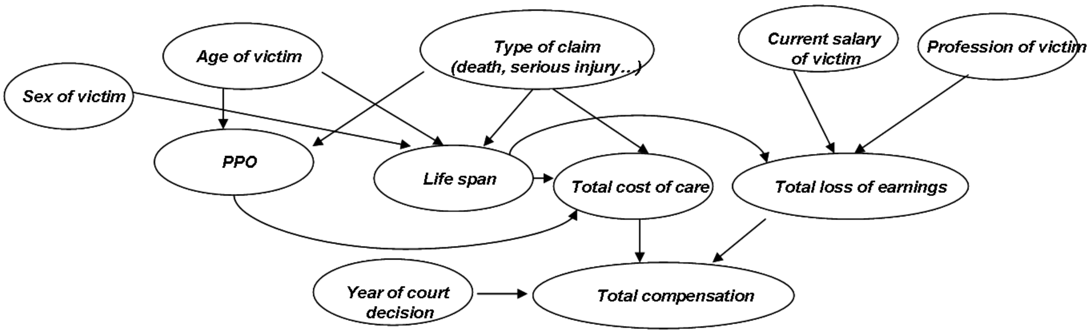


Figure 3.3: Elements that lead to the decision of granting a periodical payment instead of a lump sum in a bodily injury claim. The variable PPO is 1 if a periodical payment order has been issued by the court, 0 otherwise.

The topology of the network makes it clear that, for example, the total cost of care is affected by the type of claim (e.g. ventilated quadriplegic) and to the sex/age of the victim (but only through the life span and PPO variables!). There is a case for discarding tenuous dependencies, which are difficult to quantify anyway and therefore bring spurious accuracy.

A conditional probability table (CPT) is associated with each random variable: e.g., the life span table will be based on an annuity table for impaired lives. Total loss of earnings will depend on salary, life span and profession (which drives the likelihood of future unemployment).

3.4 Rule-based systems

The rule-based approach for dealing with uncertain knowledge is at the core of the so-called "expert systems", which were a popular product of artificial intelligence in the 1980s. Expert systems were meant to help or replace humans in making decisions in fields where there was a massive body of knowledge relevant to the decisions being taken. One popular application was in assisting doctors produce the correct diagnosis given the symptoms. The underlying methodological framework was that of mathematical logic, which provides a way to produce answers from facts and assumptions based on derivation rules.

The uncertain nature of the rules was identified from the very start, and uncertainty was dealt in a number of ways, e.g. producing alternative explanations and making use of heuristics (rules of thumb) to produce the most reasonable answer.

3.4.1 Non-monotonic logic

The results obtained with expert systems were impressive. However, the interest for them waned in the 1990s as their main flaw was realised: integrating new knowledge was fiendishly difficult and maintenance was extremely onerous, as the addition of new knowledge would sometimes lead to an overall revision of the derivation rules. This property is called *non-monotonicity*: new knowledge does not simply add on to existing one, it can sometimes make your whole body of beliefs crumble.

To make a general insurance example, consider an expert system which decides whether a motor policy customer should be insured or not. The system has a rule that if the customer is male, and is young (say in the range 17-20), then he shouldn't be insured by our company. However, if the information is then added that our customer is the happy owner of a Fiat 500 Familiare which dates back to 1967, this will make him part of a particular niche of customers that are not very risk as they are specially cautious about their treasures (as we saw when we described some results of descriptive data mining in Section 2.2.3). Therefore our expert system is too rigid and should be updated to include this exception. This seems harmless enough: however, that may compromise a number of other rules which depend on the rule on young males. (This example is not very clever. The dissatisfied reader should feel free to replace it with Tweetie's classical example, common to all books dealing with non-monotonic logics: birds fly, Tweetie is a bird, so Tweetie will be assumed to fly, but what if I added that Tweetie is a penguin?)

Non-monotonic logics [RN03] provide ways to deal with the problem of adding knowledge that defeats standard (often implicit) derivations, by modifying the notion of truth and of derivation. Two examples of non-monotonic logics are circumscription and default logic:

- **Circumscription** works by specifying predicates that are assumed to be false for every object except for those for which we know it to be true. For example, if we have the default rule that young males are dangerous ($\text{Young}(x) \wedge \text{Male}(x) \Rightarrow \text{Dangerous}(x)$), we can introduce a predicate $\text{Abnormal}(x)$ and write:

$$\text{Young}(x) \wedge \text{Male}(x) \wedge \text{Abnormal}(x) \Rightarrow \text{Dangerous}(x) \qquad (3.10)$$

Figure 3.4: Another good risk, and a headache for expert systems.

If we say that Abnormal(x) is circumscribed, that means that we can assume that x is not "abnormal" unless Abnormal(x) is known explicitly to be true. In our case, we can add the rule VintageCarOwner(x) $\Rightarrow$ Abnormal(x) by which our 500 Familiare owner is immediately exempted by the rule that (s)he must be dangerous (of course, (s)he might still be dangerous, but not because of Equation 3.10).

- **Default logic** uses default rules of the form

  $$P : J_1, \ldots J_n / C \tag{3.11}$$

  where P is called the prerequisite, C is the conclusion, and J_i are the justifications. If any of the justifications can be proven false, then the conclusion C cannot be drawn.

  In our case, the default rule will be

  $$\text{Young}(x) \wedge \text{Male}(x) : \text{Dangerous}(x) / \text{Dangerous}(x) \tag{3.12}$$

> which means that if Young(x) is true and if Dangerous(x) is consistent with the knowledge base, then Dangerous(x) may be concluded by default. The addition of the information VintageCarOwner(x) makes the predicate Dangerous(x) inconsistent with the data.

Despite much progress with non-monotonic logics has been made in recent years, there are still some unresolved questions about them. For example, it is not clear how to make decisions based on default rules, as decisions often imply weighting the strengths of different beliefs. A purely logical framework is insufficient to deal with this issue, which can be easily addressed by probability, especially in a Bayesian context.

For this reason non-monotonic logics are unlikely to be of much practical use to actuaries: as for fuzzy set theory, the approach tends to be of the logical/linguistic sort, whereas we are more interested in a quantitative approach to uncertainty.

Another approach to the addition of new knowledge which contradicts existing knowledge in a knowledge representation system is belief revision as performed by truth maintenance systems. These allow the retraction of incorrect information from knowledge bases and the removal of all inferences based on the removed information. The algorithms to perform truth maintenance are complicated and computationally demanding (NP-hard). Furthermore, it it beyond our interest for much the same reason that non-monotonic logics are. Therefore, they will not be dealt with here. The interested reader is referred to the brief section dedicated to this system in [RN03].

3.5 Practical general insurance example: Determining the parameters of a severity distribution based on estimated losses and using expert knowledge

The differences between different methods to address uncertainty are better understood through a simple example. Suppose we are pricing some type of liability business and that there is a consensus that the proper model for doing so is a single parameter Pareto distribution with cumulative distribution $F(x) = 1 - (x_0/x)^\alpha$.

The problem we want to solve is that of calculating the parameter α of the

Pareto distribution, based:

- on loss *estimates* $\{\hat{X}_1, \dots \hat{X}_n\}$ which are known with uncertainty, and also
- on prior knowledge/beliefs about the parameter (e.g. a range of possible values for α);

In the case of crisp data and no prior knowledge available, one can use MLE for a point estimate of the parameter α and calculate the standard error through the Fisher information matrix (a single value, in this case) or the bootstrap.

The MLE estimate for α is:

$$\alpha = \frac{n}{\sum_{j=1}^{n} \ln x_j - n \ln x_0} \tag{3.13}$$

When data uncertainty is present and expert knowledge is available, we need a mechanism that gives more weight to data points that are more certain (e.g. settled or almost settled claims) and that strikes a balance between what the data say and what the prior knowledge says. Let us see how this problem is solved using three different methodologies: rule-based systems, fuzzy set theory, Bayesian networks.

For illustration purposes, we consider an actual numerical example. We have generated 50 losses from a (single-parameter) Pareto distribution with threshold $x_0 = \pounds 1m$ and $\alpha = 2.5$. For each loss we have generated a random outstanding ratio (the amount yet to be paid) and we have applied distortions to each data point based on this outstanding ratio, reflecting the uncertainty on what the ultimate settlement will be, obtaining 31 losses in excess of £1m that are affected by uncertainty. We have then forgotten about how these figures were derived and we have tried to infer what the true value of α is.

We have also assumed that prior knowledge/underwriting guidelines tell us that α should be in the interval $[2, 5]$.

Table 3.1 shows the details of the artificial sample.

3.5.1 Using rule-based systems

There is naturally no unique way to solve the problem above with rule-based systems: since they employ a mixture of logic and heuristics, it all depends on

ID	Estimated loss	CV^2
1	5,603,857	0.270
2	4,556,176	0.393
3	3,763,176	0.230
4	2,673,192	0.238
5	2,256,042	0.196
6	2,062,769	0.062
7	2,016,037	0.266
8	1,958,076	0.104
9	1,934,597	0.148
10	1,882,209	0.231
11	1,778,076	0.168
12	1,741,700	0.089
13	1,669,716	0.172
14	1,570,758	0.250
15	1,521,100	0.289
16	1,499,431	0.133
17	1,463,245	0.051
18	1,440,697	0.180
19	1,440,557	0.149
20	1,412,091	0.053
21	1,357,288	0.062
22	1,252,113	0.095
23	1,235,926	0.190
24	1,194,463	0.114
25	1,193,121	0.218
26	1,191,055	0.220
27	1,190,398	0.113
28	1,164,280	0.129
29	1,158,949	0.092
30	1,103,162	0.087
31	1,041,558	0.065

Table 3.1: A sample of 31 losses randomly drawn from a single-parameter Pareto, which will be used to illustrate the different approaches to uncertainty. The uncertainty on each loss is expressed as a coefficient of variation (squared). For this particular data set, the maximum likelihood estimate of the parameter α is 1.94 (therefore lower than the 'real' value $\alpha = 2.5$). The standard error on the parameter can be calculated by standard MLE tools and turns out to be $\mathrm{se}(\alpha) = 0.35$.

which heuristics are used to take into account prior knowledge and severity uncertainty.

In the following, we give two different examples of a rule-based approach.

Rule-based approach - Example 1

The first approach is extremely simple:

1. exclude from the data set all losses whose uncertainty is greater than 50% (which implies $CV^2 > 0.25$);
2. calculate α with MLE based on the remaining data points;
3. if α_{MLE} as calculated above is between 2 and 5, then choose $\alpha^* = \alpha_{\text{MLE}}$;
4. else, if $\alpha_{\text{MLE}} < 2$, choose $\alpha^* = 2$;
5. else, choose $\alpha^* = 5$.

In our numerical example, this means that the losses with ID = 1, 2, 7, 14, 15 in Table 3.1 are discarded and MLE is applied to the remaining 26 losses, yielding $\alpha_{\text{MLE}} = 2.32 \pm 0.46$ (notice that this is closer to the original value from which values were initially drawn). Since α_{MLE} is within the admissible range $[2, 5]$, it is kept as the final result: $\alpha^* = 2.32$.

Rule-based approach - Example 2

This second example tries to elaborate a little on the previous example by introducing a credibility-weighted parameter.

1. Use MLE to get α_{MLE}, $\text{se}(\alpha_{\text{MLE}})$ and parametric bootstrap to get the standard deviation $\sigma_{BS}(\alpha)$ of different estimates of α for variations due to data uncertainty. Basically, to calculate $\sigma_{BS}(\alpha)$ one needs to calculate α_{MLE} many times for different samples based on the initial data set. Unlike non-parametric bootstrap, each sample includes exactly the same losses but the value of each loss is drawn at random from a distribution centred around the best estimate and with a standard deviation proportional to the uncertainty on the loss. This is illustrated in Table 3.2.

ID	Estim loss	CV	Sample 1	Sample 2	...	Sample 100
1	5,603,857	0.520	4,907,826	5,095,752	...	6,921,738
2	4,556,176	0.627	6,100,626		...	5,542,015
3	3,763,176	0.479	3,671,186	6,449,849	...	4,804,966
4	2,673,192	0.488	2,109,633	1,683,142	...	3,655,795
5	2,256,042	0.443	2,894,138	2,078,064	...	
6	2,062,769	0.249	1,670,715	2,138,997	...	2,165,115
7	2,016,037	0.516	1,375,701	3,187,338	...	1,445,288
8	1,958,076	0.322	2,132,508	2,153,496	...	1,761,812
...	...	...	...	...	...	...
31	1,041,558	0.255	1,110,847	1,015,721	...	1,073,547

Table 3.2: An extract of a hundred samples derived from the original dataset through parametric bootstrap. Each value of each sample is obtained by adding random Gaussian noise with coefficient of variation as in the third column to the loss in the second column. Whenever the loss falls below the threshold (£1m) the loss is replaced with a blank and does not contribute to the calculation of α (yes, this *does* introduce a bias). The standard deviation of the values of α thus found is about 0.2.

2. Use a credibility approach to combine the above with underwriters opinion: e.g., $\alpha^* = Z\alpha_{\text{MLE}} + (1 - Z)\alpha_{\text{UW}}$ (subject to α^* being between 2 and 5). In this expression, α_{UW} is the default value proposed by the underwriter, and $Z = \frac{\sigma^2_{BS}(\alpha)+se^2(\alpha_{\text{MLE}})}{\sigma^2_{BS}(\alpha)+se^2(\alpha_{\text{MLE}})+Var(\alpha_{\text{UW}})}$. $Var(\alpha_{\text{UW}})$ is the variance of the underwriters estimate

For the numerical example illustrated in Table 3.2, we have $\alpha_{\text{MLE}} = 1.94$, $\alpha_{\text{UW}} = 3.5$, se(αMLE) $= 0.46$, $\sigma_{BS}(\alpha) = 0.20$. If we assume that $Var(\alpha_{\text{UW}}) = 0.25$, this gives us an estimate of the Pareto parameter $\alpha^* = 2.72$, roughly midway between our MLE estimate and the underwriter's prior guideline.

These are just two examples of a heuristic, rule-based approach to the calculation of α. Despite the fact that these methods usually do not have a solid theoretical foundation and many different solutions are possible, reinsurers are likely to use a method along the lines of one of the examples above, in a more or less formalised fashion.

3.5.2 Using fuzzy set theory

The calculations and graphics of this section have been performed with the R package "‘fuzzyOP”, recently made available by Semagul et al. (2009).

Each non-settled loss X_i can be represented as a triangular fuzzy number $(a_1^{(i)}, a_2^{(i)}, a_3^{(i)})$, with width $a_3^{(i)} - a_1^{(i)}$ larger for losses that have a large outstanding percentage. Figure 3.6 shows examples of some of the 24 actual losses.

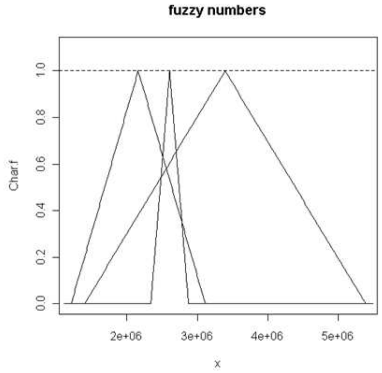


Figure 3.5: Examples of losses represented as fuzzy numbers.

Fuzzy arithmetic can be used to produce an MLE-like estimate of the parameters. E.g., for a Pareto distribution, we can use Equation 3.13.

As the logarithm is a monotonic function, it is easy to obtain the membership function of $\ln x_i$ for all losses x_i through Equation 3.3. The membership function of $\sum_{j=1}^{n} \ln x_j$ can then be obtained through Equation 3.2. This provides the membership function of the whole denominator after shifting it by $\ln x_0$. As the denominator is positive by definition (all x_i are greater than x_0)), $\frac{n}{\sum_{j=1}^{n} \ln x_j - \ln x_0}$ is a well-defined fuzzy number with finite support. The width of this support (and actually the overall shape of the membership function) is related to the uncertainty on α. Notice that this does not necessarily decrease as $n \to \infty$ (see discussion in Section 3.7).

The overall result is a fuzzy number $\hat{\alpha}$ with a membership function $\mu_{\hat{\alpha}}(x)$, whose shape can be seen in Figure 3.6 (left).

Prior knowledge can be incorporated by representing the parameter as another fuzzy number with membership function $\mu_{\alpha'}(x)$. In this experiment we

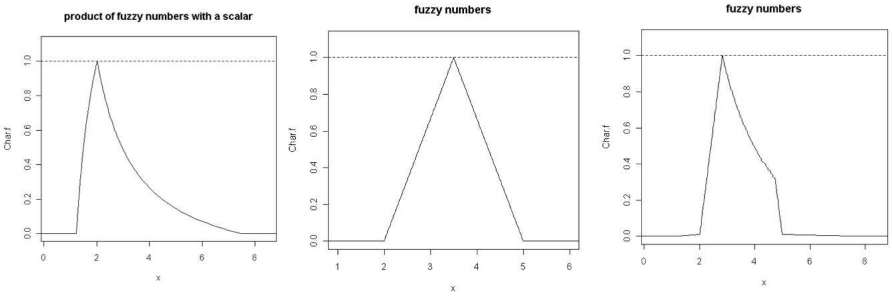


Figure 3.6: *Left:* fuzzy number resulting from the calculation of Equation 3.13. *Centre:* fuzzy number representing the prior knowledge on α (or the underwriting guidelines). *Right:* fuzzy number representing the final estimate of α, obtained as the intersection of the membership functions at the left and the right.

have modelled the prior knowledge as a triangular number between 2 and 5, centered at $\alpha = 3.5$, as in Figure 3.6 (centre).

The final result $\tilde{\alpha}$ can be obtained as the fuzzy intersection of the two estimates:

$$\mu_{\tilde{\alpha}}(x) = \mu_{\alpha' \cap \hat{\alpha}}(x) \tag{3.14}$$

and is shown in Figure 3.6 (right).

The choice of the intersection seems to make sense as it requires that both constraints (those coming from data and that coming from the soft knowledge) be satisfied at the same time.

One computational complication of the procedure is the fact that the membership function becomes more and more complex. A workaround is to ignore the finesses of the membership functions and approximate all membership functions as triangular. This will give roughly the same results and will avoid any spurious accuracy (why was the initial membership function chosen to be triangular anyway?) but it is a step towards saying that what we are doing is simply interval arithmetic with a privileged point inside the interval.

What is even more crippling, it is not clear how to extend this procedure to a case when we have a more complex distribution for which the ML solution is not analytical, e.g. with a GPD.

3.5.3 Using Bayesian analysis

In a probability context, each non-settled loss can be represented by a probability distribution centred around the best estimate of the incurred amount. The dispersion of the distribution will be related to the uncertainty regarding the loss, and this in turn depends on a number of factors, such as the outstanding ratio (outstanding amount divided by incurred amount), the size of the loss, the opinion of the claims manager on the possible variability, the age of the loss, etc.

Prior knowledge on the parameters can be incorporated by a prior distribution on the parameters themselves.

In the case where there is prior knowledge on the parameters but no data uncertainty, an estimate of the parameters θ can be obtained by maximising the posterior probability of θ given the data, that is

$$Pr(\theta|\, x_1, \dots x_n) \propto Pr(x_1, \dots x_n|\, \theta) Pr(\theta) \tag{3.15}$$

Notice that in the case where there is no prior knowledge, i.e. $Pr(\theta)$ is the non-informative prior, the term $Pr(\theta)$ can be dropped and the problem reduces to a standard maximum-likelihood problem.

Now consider the case where there is data uncertainty as well as prior knowledge: in this case the input is given by the loss estimates $(\hat{x}_1, \dots \hat{x}_n)$. For each estimate we may also have some additional information $(z_1, \dots z_n)$. For example, z_j might include the outstanding ratio and the loss date: $z_j = (\mathrm{OS}_j, t_j)$, but everything concerning the claim (e.g. the age of the victims involved, the type of injury) is relevant. For the sake of notational clarity the dependency on these variables is hidden in the equations below - but it is important to remember that this dependency exists. The posterior probability is in this case given by:

$$Pr(\theta|\hat{x}_1, \dots \hat{x}_n) = \frac{Pr(\theta)}{Pr(\hat{x}_1, \dots \hat{x}_n)} \int Pr(\hat{x}_1, \dots \hat{x}_n|\, x_1, \dots x_n) Pr(x_1, \dots x_n|\theta) dx_1 \dots dx_n \tag{3.16}$$

Equation 3.16 formally solves the problem, but it is in practice quite difficult to use without further simplifications on the structure of the problem. The problem itself is best tackled by taking a step back and using the formalism of Bayesian networks (Section 3.3). This makes use of the following standard equation:

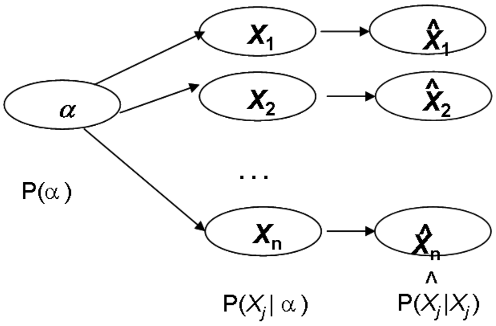


Figure 3.7: A Bayesian network representing a severity distribution with some prior knowledge on the parameters (θ) and uncertainty on the loss amounts.

$$Pr(\theta|e) = \beta Pr(\theta, e) = \beta \sum_{y} Pr(\theta, e, y) \tag{3.17}$$

where $\beta = 1/Pr(e)$, θ represents the *query variables* (in our case: the parameter(s) of the severity distribution), e is a realisation of E, representing the *evidence variables* (in our case: the estimated loss amounts), y is a realisation of Y, the *hidden variables* (in our case: the true loss amounts). Note that all of these quantities should be interpreted as vectorial random variables.

Equation 3.17 relates the posterior distribution of $Pr(\theta|e)$, which is the quantity we are interested in – as we will want to maximise this – with the joint distribution $Pr(\theta, e, y)$, which can be represented as a Bayesian network (see Section 3.3). There are many ways to specify a Bayesian networks: one which usually leads to a more compact representation is the causal model, where nodes representing causes are added first and nodes representing symptoms are added later. Diagnostic models, which do the opposite, usually result in more complicated, and less sparse, structures. In our example, a simple representation is as shown in Figure 3.7.

Some simplifications have been made in the model of Figure 3.7, by assuming that:

- the variables $\hat{x}_j$ (estimated losses) depend on α only through the variables x_k (true losses);
- $\hat{x}_j$ does not depend on x_k if $k \neq j$;

- all x_j are independent given α (a common assumption).

When one builds a model like this, a bitter discussion always ensues as to which variables can be considered to be independent of the others. In most real-world cases, there is always an argument for objecting to the claim that any two variables are independent!

Bayesian networks help introduce some discipline in this: every added link increases the complexity of the network and therefore makes inference harder, so new links should be added only if the growth in complexity is justified by the gain in accuracy, which is often spurious unless we are able to quantify these dependencies with a corresponding level of accuracy. In practice, this means that we should get rid of tenuous links (see discussion on compactness and node ordering in [RN03]).

This is particularly relevant in our example, where one cannot rigorously say, for example, that the probability of $\hat{x}_j$ does not depend on that of $\hat{x}_k$ $(k \neq j)$, as in practice the IBNER distribution is calculated based on all pairs $\{x_j, \hat{x}_j\}$.

The Bayesian network of Figure 3.7 corresponds to the following representation of the joint distribution:

$$Pr(\theta, x_1, \dots x_n, \hat{x}_1, \dots \hat{x}_n) = Pr(\theta) \prod_{j=1}^{n} Pr(x_j|\theta) Pr(\hat{x}_j|x_j) \tag{3.18}$$

Rather than $Pr(\hat{x}_j|x_j)$ it is often easier to obtain[1] $Pr(x_j|\hat{x}_j)$. By using Bayes' rule, we can re-write Equation 3.18 as

$$Pr(\theta, x_1, \dots x_n, \hat{x}_1, \dots \hat{x}_n) = Pr(\theta) \prod_{j=1}^{n} Pr(x_j|\theta) Pr(x_j|\hat{x}_j) \frac{Pr(\hat{x}_j)}{Pr(x_j)} \tag{3.19}$$

The posterior probability we are interested in is $Pr(\theta|\hat{x}_1, \dots \hat{x}_n)$, which can be written by using Equation 3.17 as

$$Pr(\theta|\hat{x}_1, \dots \hat{x}_n) = \frac{1}{\prod_{j=1}^{n} Pr(\hat{x}_j)} \int Pr(\theta, x_1, \dots x_n, \hat{x}_1, \dots \hat{x}_n) dx_1 \dots x_n \tag{3.20}$$

[1] Notice that our Bayesian network could be re-arranged by putting a direct link from $\hat{x}_j$ to x_j (rather than vice versa), turning it into a diagnostic – rather than causal – network.

After some very limited algebra, Equation 3.20 reduces to Equation

$$Pr(\theta|\hat{x}_1, \dots \hat{x}_n) = Pr(\theta) \prod_{j=1}^{n} \int \frac{Pr(x|\hat{x}_j)Pr(x|\theta)}{\int Pr(x|\theta')Pr(\theta')d\theta'} dx \tag{3.21}$$

Notice that if there is no data uncertainty, $Pr(x|\hat{x}_j)$ reduces to Dirac's delta distribution: $Pr(x|\hat{x}_j) = \delta(x - \hat{x}_j)$ (informally, a distribution concentrated at point $\hat{x}_j$ of zero width and infinite density) and Equation 3.21 reduces to

$$Pr(\theta|\hat{x}_1, \dots \hat{x}_n) = Pr(\theta) \prod_{j=1}^{n} \frac{Pr(\hat{x}_j|\theta)}{\int Pr(\hat{x}_j|\theta')Pr(\theta')d\theta'}$$

which is nothing but Equation 3.15 with the incorporation of the independence assumption of the variables $\hat{x}_j|\theta$.

Despite the complex form of Equation 3.21, this is a function of a (vector) parameter θ, which we can maximise by numerical methods, e.g. by Markov chain Monte Carlo methods. In the case where θ is a scalar (e.g. the Pareto distribution assumed for the fuzzy case) the problem can be solved quite efficiently.

More specifically, in our example there is a single parameter according to which we have to maximise the posterior probability, the exponent of a single-parameter distribution. In order to use a standard notation we therefore set $\theta = \alpha$. The three functions we need to know to calculate the integrals in Equation 3.21 above are:

- the conditional probability $Pr(x|\alpha)$. For a Pareto distribution this is

  $$Pr(x|\alpha) = \alpha \frac{x_0^{\alpha}}{x^{\alpha+1}} \tag{3.22}$$

  where x_0 is the threshold above which the distribution is defined. The parameter α must be greater than 1 for the mean of the distribution to be defined, and the larger it is, the quicker the tail of the distribution dies off;

- the prior probability $Pr(\alpha)$. This is obviously a delicate choice for all Bayesian analyses and should reflect our prior knowledge on the parameters - e.g. that the Pareto's parameter α must be between 2 and 5. For example, we might assume that the prior distribution of α is a Beta distribution between 2 and 5:

$$Pr(\alpha) = c \cdot (\alpha - 2)^{a-1} \cdot (5 - \alpha)^{b-1}, \tag{3.23}$$

where a and b (both > 1) are such that the mean of the distribution is 3.5 (the preferred underwriter's choice). Here are some properties of the Beta distribution above, which are also illustrated in Figure 3.8.

- If $a = b$, the distribution has a mode in the middle of the interval, in this case $\alpha = 3.5$ (which ties in with our underwriter's best estimate) and is symmetric with respect to the mode. The absolute value of a $(= b)$ then gives the width of the distribution: the higher a is, the narrower the distribution. Notice that for $a <= 2$, the distribution will be fully concave, whereas for $a > 2$ there will be two changes of convexity. Figure 3.8 shows the Beta distribution for three different values of a.
- If $a < b$, the distribution is skewed to the left, if $a > b$ it is skewed to the right.

The exact shape of the distribution is actually not important: there is no way that we can have detailed information on the nuances of the prior distribution! What is important is to understand how the properties described above can be used to incorporate prior information.

- the conditional probability $Pr(x|\hat{x})$, or rather (to go back to the complete notation) $Pr(x|\hat{x}_j, z_j)$. This is nothing but the IBNER distribution, and can be chosen to reflect our belief on the accuracy of each estimate, but it should normally be informed by the actual reserving experience. Although the experience may be limited (especially if one wishes to consider the effect of the outstanding ratio, the loss date, etc.) it will often be possible to have a mean IBNER factor and a variance. Given an acceptable model for the distribution (e.g. Gamma, which is non-zero only for positive losses), it is then possible to create an adequate model for $Pr(x|\hat{x})$. For example, we might use for $Pr(x|\hat{x}_j)$ a Gamma distribution

$$Pr(x|\hat{x}_j, z_j) = c' x^{u(\hat{x}_j, z_j)-1} \exp(-x/v(\hat{x}_j, z_j)), \tag{3.24}$$

where u and v are functions of $\hat{x}_j, z_j$ and are chosen so that the mean of x is $E(x) = u \cdot v = \hat{x}_j$ and the variance $\mathrm{Var}(x) = u \cdot v^2$ is equal to the empirical variance $\mathrm{Var}(\hat{x}_j)$, leading to:

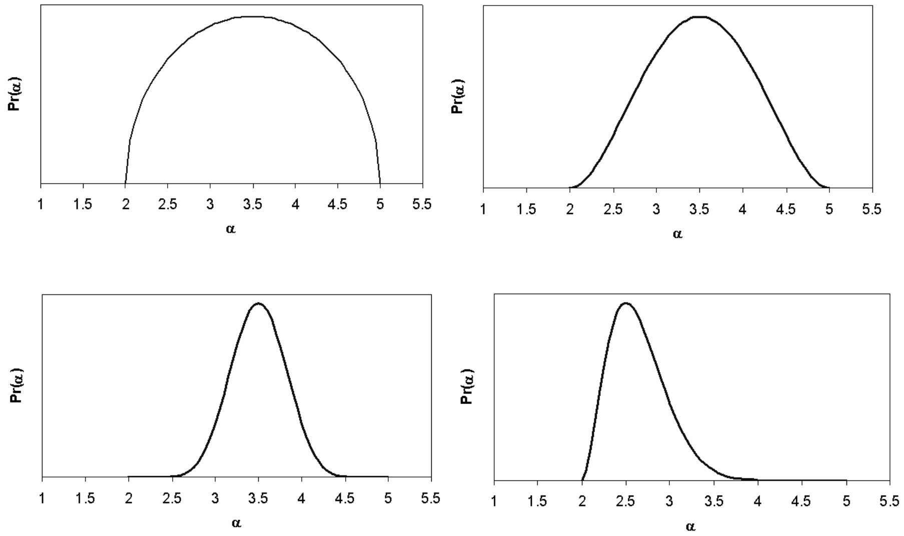


Figure 3.8: Shape of the Beta distribution for different values of the parameters a, b. *Top left*: $a = b = 1.5$. Notice how in this case the distribution is always concave. *Top right*: $a = b = 3$. In this case the distribution changes convexity twice. The standard deviation of this distribution is quite large, signalling that we consider the best estimate for α only a broad indication. *Bottom left*: $a = b = 11$. This is as the top right case but narrower – signalling that we believe that it is unlikely that the true value of α should depart much from the underwriting guideline. *Bottom right*: $a = 2$, $b = 11$. In this case the distribution is skewed to the left. In many circumstances, a skewed distribution will be more suitable, as there might well be more room for variability on one side than on the other.

$$u = \hat{x}_j / CV^2(\hat{x}_j, z_j), v = CV^2(\hat{x}_j, z_j) \tag{3.25}$$

where CV is the coefficient of variation.

A simple alternative is using a Gaussian distribution:

$$Pr(x|\hat{x}_j, z_j) = \frac{1}{\sqrt{2\pi}\sigma_j} \exp(-\frac{(x - \hat{x}_j)^2}{2\sigma_j^2}) \tag{3.26}$$

where $\sigma_j = CV(\hat{x}_j, z_j)\hat{x}_j$. This will give acceptable results when the coefficient of variation is not too large (and therefore it will be unlikely to produce negative losses) and when the distribution is reasonably symmetric.

> This approach is usually called **empirical Bayesian approach**, as the prior distribution is actually estimated based on some previous, or external, experience, rather than made up from scratch[2].

By substituting the expressions for $Pr(x|\alpha)$ and $Pr(\alpha)$ into Equation 3.21, we obtain (for each element of the product):

$$\begin{aligned} g_j(\alpha) &= \int \frac{Pr(x|\hat{x}_j)Pr(x|\alpha)}{\int Pr(x|\alpha')Pr(\alpha')d\alpha'}dx \\ &= \alpha \int_{x_0}^{\infty} I^{-1}(x/x_0)Pr(x|\hat{x}_j)e^{-\alpha \ln(x/x_0)}dx \end{aligned} \tag{3.27}$$

where

$$I(t) = \int_2^5 \alpha(\alpha-2)^{a-1}(5-\alpha)^{b-1}e^{-\alpha \ln t}d\alpha \tag{3.28}$$

Equations 3.27 and 3.28 are not for the seeker of mathematical beauty and simplicity – however, they allow us to reformulate the original problem as the problem of finding the value of α which maximises the product of a number of factors of the form above:

$$\alpha^* = \text{argmax}\left\{(\alpha-2)^{a-1}(5-\alpha)^{b-1}\prod_{j=1}^{n} g_j(\alpha)\right\} \tag{3.29}$$

Solving Equation 3.29 is an elementary problem of numerical optimisation in one variable (α). The only additional difficulty is that the function $I(t)$ is not available analytically but needs to be calculated by numerical integration itself for different values of t. Figure 3.9 depicts the behaviour of function $I(t)$ for $t \geq 1$.

The final result – the posterior distribution for α for the dataset illustrated at the beginning of Section 3.5 – is shown in Figure 3.10

[2] Notice that we have glossed over the issue of losses becoming larger or smaller than the threshold x_0 as a side-effect of data uncertainty. This is of course an important practical aspect but would be a distraction here.

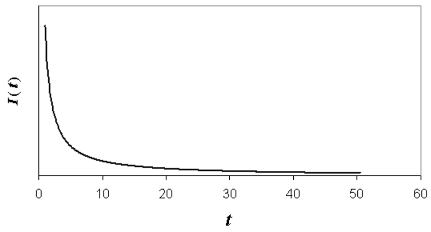


Figure 3.9: The behaviour of function $I(t)$ for $t \geq 1$.

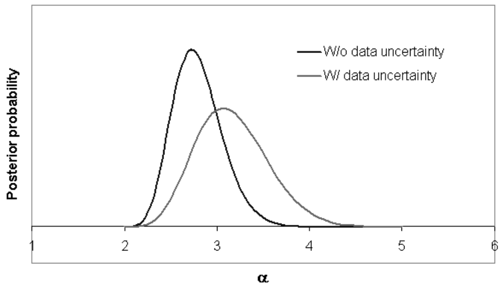


Figure 3.10: The posterior distribution for α for our example dataset, with and without data uncertainty. A Beta distribution with α between 2 and 5 and $a = b = 6$ has been used as a prior, and a Gaussian data uncertainty distribution (Equation 3.26 has been used. In this case the distribution with data uncertainty is wider and is also closer to the underwriter's guideline ($\alpha = 3.5$), probably because data uncertainty generally increases the uncertainty on the parameter calculated from the data.

3.6 Practical general insurance example: Model uncertainty

As mentioned in the introduction to this section, model uncertainty is a particularly thorny issue as the problem of quantifying model uncertainty is as difficult as that of determining what the correct model is – or equivalently, what the correct theory describing a phenomenon is! What was the model uncertainty of Newton's theory of gravitation? One can only attempt to reply to these questions only for nested theories – for example, it may be possible to find some measure of how good Newton's theory of gravitation is as an approximation of Einstein's general relativity. But what about the model uncertainty of general relativity itself, which hasn't yet been replaced by a more powerful theory?

A dumbed-down version of this problem, however, seems to be more approachable: this is the case where there is only a finite number of competing models available, and we want to select the one that is better at predicting the results on an independent sample.

In this case we can assign to each model a prior probability – for example the same probability, if there is no other reason to do otherwise – and calculate the posterior probability of the model given the data. This is explored for example in [HTF01] and, to a greater extent, in [Bis07].

The main idea is to compare a set of L models $\{\mathbf{M}_i\}$ where $i = 1, \dots L$. Each model is a probability distribution. Each model has a prior probability $\Pr(\mathbf{M}_i)$. Given a training set $D = \{X_1, \dots X_n\}$, we are interested in evaluating the posterior distribution

$$\Pr(\mathbf{M}_i|D) \propto \Pr(\mathbf{M}_i)\Pr(D|\mathbf{M}_i)$$

(3.30)

In the equation above,

- the term $\Pr(\mathbf{M}_i)$ expresses a preference for different models;
- the term $\Pr(D|\mathbf{M}_i)$ expresses the preference shown by the data themselves for different models[3], and is called the *model evidence.*

Equation 3.31 is all we need in those contexts where we simply want to select the best among a number of competing models. After some elaborations, it can also be used as an approximate analytical criterion for model validation, called Bayesian Information Criterion (see Section 2.1.2). A derivation of this criterion from Equation 3.31 can be found in [Bis07].

Going back to our example of Section 3.5, we might be undecided as to whether we want to model our losses with a Pareto (thick tail) or an exponential distribution (thin tail), and we might have a preference for Pareto,

[3] The ratio of model evidences for two models, $\Pr(D|\mathbf{M}_i)/\Pr(D|\mathbf{M}_j)$ is called the Bayes factor.

because (for example) for 70% of our clients the Pareto distribution turned out to be more successful in fitting our data. Therefore we might assign a 70% prior probability to the Pareto model and a 30% probability to the Exponential model (again, this is an empirical Bayesian approach!).

The posterior probability of the Pareto distribution is now given by:

$$\Pr(\mathbf{M}_{\text{Pareto}}|D) \propto \Pr(\mathbf{M}_{\text{Pareto}})\Pr(D|\mathbf{M}_i) \tag{3.31}$$

and if this is larger than 0.5 we may choose a Pareto model (as opposed to an exponential) for modelling our data.

3.7 Comparing the Bayesian and the fuzzy logic approach

Several authors have compared the Bayesian and the fuzzy logic approach to uncertainty. This debate has spread to the actuarial community – for a taster of this debate, see the interesting paper by Cummins and Derrig [CD97], especially the discussion between Van Slyke and the authors in the final section of the paper. Buckley [Buc83] has compared fuzzy set theory and the probabilistic approach, abstaining from drawing any firm conclusions on what method is best but identifying areas where one may be preferred to the other. The paper [NCC99] sheds some light on how fuzzy set theory differs from probabilistic methods when averaging over many variables.

The wrong type of uncertainty? The most common criticism of fuzzy set theory is probably that it is nothing but an unwieldy version of probability theory, and that everything that can be dealt with by fuzzy set theory can also be dealt with by Bayesian probability. Is it fair to say so? It is true that fuzzy set theory and probability theory have much in common: both deal with set theory, and both can represent uncertainty and soft knowledge. When it comes to representing numbers, membership functions look very much like distribution probabilities – so why not just using those? The classical response to this is that fuzzy set theory captures fuzzy membership to a set, whereas probability distributions aim at calculating the likelihood that an element belongs to a set in a crisp way. The conceptual difference is indeed important: when I say for example that a loss is large, I am using a fuzzy concept. No probability is involved here - the loss may be exactly £130,000 and may have already been settled - but whether this loss belongs to the set of large losses is indeed fuzzy (and context-dependent).

My view on this debate is that: (a) it is of course true that fuzzy set theory and probability theory are two conceptually distinct disciplines, each modelling different aspects of uncertainty. The notion of a fuzzy set is a very natural development of set theory and it is difficult to imagine set theory without it now! Having said that, (b) the interest of fuzzy set theory seems to me to be mostly logical and linguistic. It may be useful for modelling the functionality of sensors based on vague inputs, but in a highly numerical context like that of financial applications, it is indeed not clear what fuzzy logic can do that probability theory cannot do better. The type of uncertainty we are interested in is not of a linguistic kind: we may not be able to define crisply what a "large" loss is, but there's hardly a business case for attempting it! Rather, we often have to deal with a situation where the amount of the loss is indeed uncertain, and we might have some prior knowledge on what the amount might be (something that a Bayesian prior will capture easily, at least from a formal point of view). In other words, the uncertainty we are dealing with day in, day out is an uncertainty on the amounts, not a semantic uncertainty! I would go as far as to say that in all of the applications in the actuarial literature discussed in Section 3.1.1 it is this second type of uncertainty which is addressed – for example, it is so in the case of the paper by Cummins and Derrig [CD97], where fuzzy logic is used to quantify the uncertainty on the risk-adjusted discount rate for losses, on the losses themselves, on the premiums, etc.

Is fuzzy set theory unwieldy? As mentioned above, many authors maintain that the mathematical tools for dealing with fuzzy objects is far less developed than probability theory. This is a fair point, as one of the main reasons of the recent success of Bayesian analysis lies in the fact that many tools have been developed for the calculation of posterior distributions: these include Markov Chain Monte Carlo techniques, Gibbs sampling, Metropolis-Hastings, and all the techniques applicable to Bayesian networks to take advantage of their sparsity. On the other hand, everything beyond simple arithmetic is painful with fuzzy set theory. Until a proper calculus is developed, only elementary applications to general insurance will be possible.

Parameter uncertainty. Another issue with fuzzy set theory is that it deals poorly with parameter and model uncertainty. As observed for example in [NCC99], the law of large numbers does not hold for fuzzy numbers. The average of n fuzzy numbers with the same membership function has *exactly* the same membership function of the component numbers, regardless of n. On the contrary, in a probabilistic setting the average of n independent numbers will have a narrower distribution than its components, and this narrowing can be rigorously quantified by the central limit theorem. This leads to the apparent paradox when dealing with uncertain knowledge that adding more knowledge does not lead to a reduction in uncertainty! The problem lies of course in the fact that the extension principle for fuzzy arithmetic does not address the issue of whether two fuzzy numbers are independent or not (and what does “independence” mean anyway in the context of fuzzy set theory?). The sum of fuzzy numbers can be likened to the sum of perfectly correlated variables.

Model uncertainty. There is no obvious way by which fuzzy set theory can deal with model uncertainty. In [CD97], one of the stated reasons to use fuzzy set theory was actually that there is uncertainty as to whether capital asset pricing model is the correct model to produce a risk-adjusted discount rate for losses, r_L (see [CD97], Section 2.3). However, this fact in only used to justify an additional uncertainty on the result of this model – r_L itself. In other words, this model uncertainty is translated into a sort of parameter uncertainty. On the contrary, Bayesian theory incorporates model uncertainty by using the prior probability on models, and gives a recipe on how to choose between a number of competing models. It is certainly a limited solution but it is more far-reaching than what one can do with fuzzy set theory.

Other issues. As discussed for example in [Ben96], fuzzy set theory has problems in applications where multiple levels of operations and decisions must be made. Furthermore, some of the basic definitions of fuzzy set theory, such as the definition of intersection, lead to counterintuitive results and need to be amended *ad hoc* depending on the application [Lem90].

Chapter 4

The temporal aspects of risk

FIRST LAW: Loss data from the last 4 years are not sufficiently developed to be credible.
SECOND LAW: Loss data older than 4 four years are not relevant anymore.
THIRD LAW: You need at least 10 years of credible, relevant data to do experience rating.
FOURTH LAW: The past is not a good guide to the future.

The four laws of pricing long-tail business. (London Market anonymous)

Risk has a temporal aspect in at least two ways:

1. evidence about risk (which is then used for pricing or reserving purposes) unfolds gradually: in some cases, such as in long-tail liability classes, it may take two decades or more to know for certain what the losses for a given underwriting year were;

2. risk itself is ever-changing, which makes the evidence collected in the past gradually obsolete. This second issue is of course thornier than the first one, as the useful assumptions of stationarity on which many of the techniques for interpreting data rely do not hold.

One obvious example from general insurance that comes to mind when thinking about the temporal element is experience rating. In order to calculate the risk premium for a policy, claims data are collected over a number of years,and a frequency and severity distribution are calculated based on the available experience. This is usually done statically – the experience accumulated over, say, 10 years is put in a single pot and analysed to find the best estimate of the risk premium. However, this can be viewed dynamically by considering how the estimate of the risk premium changes year-on-year based on new evidence. This reflects how premiums are set in reality – starting from an initial guess and changing them as we get wiser.

Another obvious example is reserving – reserves are changed periodically in the light of new evidence, with an underlying model of change in mind.

When stationarity holds – the environment does not change and the risk does not change over the years – we expect to obtain the same results whether we look at data statically or dynamically. However, when the environment constantly changes we have the additional problem of how to weigh past experience against more recent evidence. New evidence allows us to refine parameter calculation but may well force us to change the parameters themselves or even the models! This is a problem faced by underwriters of commercial lines (or of reinsurance) constantly, as clients keep changing their risk control mechanisms and expect to see the price of their policy reflect this.

Several approaches have been used in computational intelligence to model time dependence. We are going to illustrate three of them: hidden Markov models (HMM's), Kalman filtering and dynamic Bayesian networks. Before describing these approaches, we need to give some definitions and explain some generalities.

4.1 Definitions and generalities

Temporal aspects are usually addressed in computational intelligence by using a probabilistic temporal model. Some authors (see for example [JZ83]) call this a state-space model.

Some simplifications are usually made: the underlying phenomenon is regarded as both stationary and having the Markov property.

- Stationarity: the change is governed by laws that do not themselves change over time. Notice the distinction between stationary (the process is ruled by known, unchanging laws) and static (the state of the system does not change).

- Markov assumption: the current state depends only on a finite history of previous states (which can be reduced to one – a first-order Markov process).

A probabilistic temporal model needs three ingredients:

A transition model. This is the conditional distribution $\mathbf{P(X_t|X_{t-1})}$, and fully describes time evolution in the case of first-order Markov processes.

A sensor model (aka observation model). This is the probability $\mathbf{P(E_t|X_t)}$, describing how the evidence is affected by the state of the world. We usually assume that the evidence only depends on the current state, i.e. $\mathbf{P(E_t|X_{0:t}, E_{0:t-1}) = P(E_t|X_t)}$. The state of the world is supposed to be "hidden", that is, measurable only through the observation model.

A prior probability. This is the probability $\mathbf{P(X_0)}$ over the states at time zero.

These ingredients can be combined to specify the whole joint distribution:

$$\mathbf{Pr(X_0, X_1, \ldots X_t; E_1, \ldots E_t) = Pr(X_0) \prod_{i=1}^{t} Pr(X_i|X_{i-1}) Pr(E_i|X_i)} \quad (4.1)$$

4.2 What can temporal models do?

We have now introduced temporal models and given some basic definitions: what shall we do with them? Examples of inference tasks that can be achieved with temporal models are listed below. This list is borrowed from [RN03] and put in the context of general insurance.

Monitoring (or filtering): i.e., calculating the belief state (the posterior distribution $Pr(X_t|e_{1:t})$ of the current state given all the evidence to date). For example, X_t may represent the number of losses expected in year t given all claims reported up to t. The monitoring task can be solved recursively as follows

$$Pr(X_{t+1}|e_{1:t+1}) = \alpha Pr(e_{t+1}|X_{t+1}) \sum_{x_t} Pr(X_{t+1}|x_t) Pr(x_t|e_{1:t}) \quad (4.2)$$

Equation 4.2 requires the application of Bayes rule and of the Markov property of evidence (see Section 4.1). The term $Pr(X_{t+1}|x_t)$ in Equation 4.2 is the transition model and $Pr(x_t|e_{1:t})$ is the current state distribution.

Predicting: i.e., calculating the posterior distribution of a future state given the evidence until now, $Pr(X_{t+k}|e_{1:t})$. A typical example is experience rating – the calculation of the future distribution of claims given historical losses (experience rating).

Note that prediction can be seen as monitoring without the benefit of new evidence. The recursive equation for prediction can be written as:

$$Pr(X_{t+k+1}|e_{1:t}) = \sum_{x_t} Pr(X_{t+1}|x_{t+k}) Pr(x_{t+k}|e_{1:t}) \quad (4.3)$$

Smoothing (or hindsight): calculating the posterior distribution of a past state given all evidence up to the present, $Pr(X_k|e_{1:t})$ $(k < t)$. A classical example is reserving, where we need to keep updating our estimate of losses in year k for many years after that – until we are fully sure that no more claims can be reported for that period (asbestosis claims are an example which shows the relevance of this).

Notice that the transition and the observation model themselves must often be learned from observations.

4.3 Hidden Markov models

A hidden Markov model (HMM) is a temporal probabilistic model, in which the state of the process is described by a single discrete random variable (possibly, a vector variable). This representation allows a simple and elegant matrix implementation of the generic model [RN03].

The transition model is represented as

$$\mathbf{T}_{i,j} = Pr(X_t = j|X_{t-1} = i) \quad (4.4)$$

The sensor model is represented as the matrix

$$\mathbf{O}_t = \mathrm{diag}(Pr(e_t|X_t = i)) \tag{4.5}$$

(that is, a diagonal matrix whose diagonal elements are given by the values of $Pr(e_t|X_t = i)$.

The updating equation is

$$\mathbf{f}_{1:t+1} = \alpha \mathbf{f}_{t+1} \mathbf{T}^T \mathbf{f}_{1:t} \tag{4.6}$$

Where $\mathbf{f}_{1:t} = Pr(x_t|e_{1:t})$ is the so-called 'forward message'.

4.4 Kalman filtering

Kalman filtering is a type of regression analysis enhanced with a mechanism for updating the regression parameters, which are assumed to evolve in time. A classical application of Kalman filtering is radar tracking: once an object is identified at time t, its position at time $t + \Delta$ is determined by combining the observation received at time $t + \Delta$ (which may be noisy and therefore misleading – this is probably easier to imagine if you consider radars in World War II) with an assumption on where the object is expected to be found at time $t + \Delta$ based on its position and speed at time t.

Although various extensions of the Kalman filter can be found in the literature, the standard model of Kalman filtering requires the key assumption that the current state follows a multivariate Gaussian distribution. The transition model for the Kalman filter is therefore:

$$Pr(\mathbf{x_{t+1}}|\mathbf{x_t}) = \mathbf{N}(\mathbf{F}\mathbf{x_t}, \mathbf{\Sigma_x})(\mathbf{x_{t+1}}) \tag{4.7}$$

whereas the sensor model is

$$Pr(\mathbf{z_t}|\mathbf{x_t}) = \mathbf{N}(\mathbf{H}\mathbf{x_t}, \mathbf{\Sigma_z})(\mathbf{z_t}) \tag{4.8}$$

In the equations above,

- $\mathbf{F}$ and $\mathbf{\Sigma_x}$ are matrices describing the linear transition model and transition noise covariance;

- $\mathbf{H}$ and $\mathbf{\Sigma_z}$ are matrices describing the linear sensor model and sensor noise covariance.

The update equation for the mean μ_t of $\mathbf{x}_t$ is now:

$$\boldsymbol{\mu}_{\mathbf{t+1}} = \mathbf{F}\boldsymbol{\mu}_{\mathbf{t}} + \mathbf{K_{t+1}}(\mathbf{z_{t+1}} - \mathbf{HF}\boldsymbol{\mu}_{\mathbf{t}}) \tag{4.9}$$

and the update equation for the covariance is:

$$\mathbf{\Sigma_{t+1}} = (\mathbf{I} - \mathbf{K_{t+1}})(\mathbf{F\Sigma_t F^T} + \mathbf{\Sigma_x}) \tag{4.10}$$

where $\mathbf{K_{t+1}}$ (the Kalman gain) is given by:

$$\mathbf{K_{t+1}} = (\mathbf{F\Sigma_t F^T} + \mathbf{\Sigma_x})\mathbf{H^T}(\mathbf{H}(\mathbf{F\Sigma_t F^T} + \mathbf{\Sigma_x})\mathbf{H^T} + \mathbf{\Sigma_z})^{\mathbf{-1}} \tag{4.11}$$

Despite their “hairy horribleness” (to borrow Russell and Norvig’s expression), a few moments devoted to contemplating Equations 4.9, 4.10, 4.11 will remind the actuary of credibility theory: the Kalman gain is the credibility given to the new observation $\mathbf{z_{t+1}}$ (or rather, the difference between this and the predicted observation $\mathbf{HF}\boldsymbol{\mu}_{\mathbf{t}}$), relative to the predicted state at time $t+1$, $\mathbf{F}\boldsymbol{\mu}_{\mathbf{t}}$.

As mentioned at the beginning of this section, there have been attempts to overcome the strong assumption of a linear Gaussian transition and sensor model:

- the **extended Kalman filter** (EKF) is an attempt to incorporate nonlinearities by considering a model which is locally linear. This works well for smooth, well-behaved systems – that is, systems that do not have abrupt changes;

- the **switching Kalman filter**, which attempts to model strongly nonlinear behaviour such as obstacle avoidance by using multiple Kalman filters among which it is possible to switch.

Both cases above are special cases of dynamic Bayesian networks, that provide a far more general solution to the problems addressed by the Kalman filter.

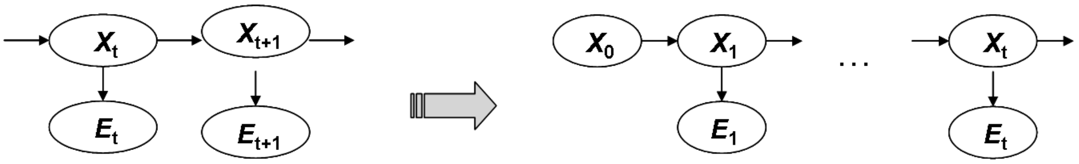


Figure 4.1: A dynamic Bayesian network, in recursive (left) and unrolled (right) form

4.5 Dynamic Bayesian networks

Dynamic Bayesian networks are Bayesian networks representing temporal models.

They are completely general in the sense that they do not add any further assumptions to those of general temporal models, i.e. stationarity and the Markov property.

Complete joint distribution:

$$\mathbf{Pr(X_0, X_1, \ldots X_t, E_1, \ldots E_t) = Pr(X_0) \prod_j Pr(X_j|X_{j-1}) Pr(E_j|X_j)} \tag{4.12}$$

This is all we need to solve the prediction problem:

$$\mathbf{Pr(X_{t+1}|e_{1:t+1})} = \alpha \mathbf{Pr(e_{t+1}|X_{t+1}) \Sigma_{x_t} Pr(X_{t+1}|x_t) Pr(x_t|e_{1:t})} \tag{4.13}$$

A representation of a dynamic Bayesian network in an unrolled/rolled-up version is shown in Figure 4.1. The rolled-up version is that where only the recursive section is shown; the unrolling refers to the development of the recursion.

Note that a Kalman filter can always be represented as a dynamic Bayesian network, as illustrated in Figure 4.2.

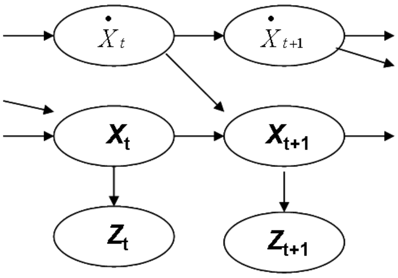


Figure 4.2: Representation of a Kalman filter (depicting a classical situation where both the position (x_t) and the speed ($\dot{x}_t$) of an object are tracked) as a dynamic Bayesian network.

Origin year, s	Inflation index	Volume index	0	1	2	3	4
1	0.598	1.000	753.5	648.9	311.7	173.5	71.3
2	0.665	0.899	642.3	648.4	249.7	206.5	
3	0.748	0.858	715.8	661.1	309.4		
4	0.853	0.863	841.6	862.6			
5	1.000	0.813	968.8				

Table 4.1: Example of a run-off triangle. The amounts in the triangle represent the incremental paid amounts for each origin year and for each development year (0 to 4). Amounts shown are in £1,000.

4.6 Practical general insurance example: Reserving

One well-known application of probabilistic temporal models to general insurance is that of De Jong and Zehnwirth [JZ83] to reserving. The general case is quite complex so for illustrative purposes we will focus on the simple example that can be found in Section 5 of their paper.

This is based on the run-off triangle in Table 4.1 (borrowed from [JZ83]), that gives the amount paid for each development year and for each origin year, taking into account some measure of inflation (the inflation index) and of exposure (the volume index).

In this example, the observation (evidence) variables $\mathbf{z_t}$ are the successive diagonals of the paid triangle, as they represent the evidence collected in the latest calendar year. In the example above,

$$\begin{aligned}
\mathbf{z_1} &= (753.5)^T \\
\mathbf{z_2} &= (642.3, 648.9)^T \\
\mathbf{z_3} &= (715.8, 648.4, 311.7)^T \\
\mathbf{z_4} &= (841.6, 661.1, 249.6, 173.5)^T \\
\mathbf{z_5} &= (968.8, 862.6, 309.4, 206.5, 71.3)^T
\end{aligned}$$

The underlying parameters are in this simple case the ultimate incurred claims x_s for each origin year s. The **sensor equation** (or observation equation, which is more suitable to this context), which connects the paid amounts to the ultimate claims x_s is (in this extremely simplified model):

$$\begin{aligned}
z_t(d) &= n(t-d)\cdot\lambda(t)\cdot x_{t-d}\cdot\phi(d)+u_{t-d} \\
&= n(t-d)\cdot\lambda(t)\cdot x_{t-d}\cdot(d+1)\cdot\exp(-d)+u_{t-d}
\end{aligned}$$

In the equation above, u_{t-d} is a zero-mean random noise and $x_{t-d}\cdot(d+1)\cdot\exp(-d)$ is the expected value of payments for origin year $t-d$ and development year d, which is decreasing monotonically as a function of the development year. In this setting, the unknown parameter is x_{t-d}, which gives the level of the exponentially decreasing payments.

The transition model is given by the simple random walk model:

$$x_s = x_{s-1} + v_s \qquad (4.14)$$

where v_s is again a zero-mean random noise.

We also need an initial state x_0 corresponding to the *unobserved* origin year $s = 0$.

In [JZ83], this problem is approached by using extended Kalman filtering, with the structural parameters ($\mathbf{F}$, $\mathbf{H}$,...) varying with time and therefore indicated below as $\mathbf{F}(t)$, etc. The list below explains what these parameters look like in this simple case:

- the matrix $\mathbf{F}(t)$ is a matrix with zeros everywhere except at elements $(1,1)$, $(2,1)$, $(3,2)$, ... $(t,t-1)$ where it is 1;

			Time		
	1	2	3	4	5
x_5					1147.3
					(53.7)
x_4				1141.2	1146.2
				(55.9)	(48.9)
x_3			1156.4	1141.1	1140.6
			(57.2)	(51.2)	(45.3)
x_2		1157.5	1157.1	1140.2	1138.5
		(57.2)	(52.3)	(47.3)	(42.8)
x_1	1154.0	1158.6	1156.9	1142.5	1139.4
	(55.4)	(52.2)	(48.2)	(44.4)	(41.2)

Table 4.2: The estimates of the states x_t at different times. The quantities shown between brackets are the standard errors.

- the matrix $\Sigma_x(t)$ is written as $G(t)V(t)G^T(t)$ where $G(t)$ is a $t \times 1$ matrix whose only non-zero element is $(1,1)$ which can be set to 1, and $V(t)$ was estimated by MLE methods as $V(t) = 626.3$;

- the initial state x_0 corresponding to origin year $s = 0$ and its covariance matrix Σ_0 are back-calculated with MLE methods assuming that a number of diagonals are already given, obtaining the values $x_0 = 1146.3$ and $\Sigma_0 = 2667.3$;

- the covariance matrix $\Sigma_z(t)$ is a diagonal matrix whose diagonal values are the first t values from $\text{diag}(16283.6, 12204.8, 4509.3, 2755.7, 876.8)$

- Kalman's update rules (Section 4.4) are then used to produce the projected values of the claims triangle.

The estimates for the states x_t generated by the Kalman filter are shown in Table 4.2

The results above could be also obtained by a dynamic Bayesian network like that in Figure 4.3. The model can easily be enhanced by adding nodes which represent (possibly discrete) random variables W_t incorporating prior knowledge on the things that might have an impact on the general level of claims for a given origin year (for example, a bad winter, or the introduction of new driving legislation) or on the amounts paid from a given year on (for example, new legislation on court awards for bodily injury liability). The transition model and the sensor model including these additional variable need not of course be linear or be Gaussian.

Note that the model shown in Figure ?? will require some rearrangement before being cast as a standard transition/sensor model, since X_t affects Z_t, Z_{t+1}, ... Z_{t+k} (k is the maximum number of development periods such that the hidden variables X_t are relevant to the evidence variables: in other words, we can assume that $Pr(Z_{t+j}|X_t, \Omega_t) = Pr(Z_{t+j}|\Omega_t)$ for $j > k$). However, as long as k is finite, this process can always be transformed into a first-order Markov process by a suitable redefinition of variables.

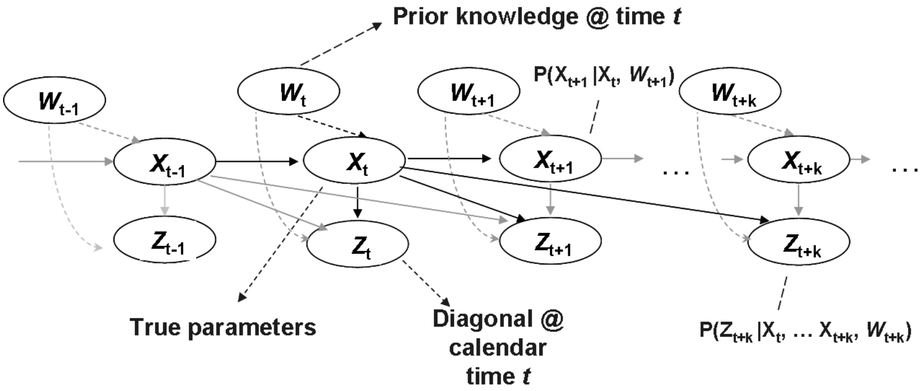

Prior knowledge @ time t
$P(X_{t+1}|X_t, W_{t+1})$
W_{t-1}
W_t
W_{t+1}
W_{t+k}
X_{t-1}
X_t
X_{t+1}
...
X_{t+k}
...
Z_{t-1}
Z_t
Z_{t+1}
Z_{t+k}
True parameters
Diagonal @ calendar time t
$P(Z_{t+k}|X_t, \ldots X_{t+k}, W_{t+k})$

Figure 4.3: A dynamic Bayesian network for loss reserving. The example in [JZ83] does not require the variables W_t, depicting prior knowledge, which are therefore connected to the rest of the network with dashed lines. All the edges except those to and from X_t are grayed.

4.7 Comparison of temporal models

In this chapter we have looked at probabilistic reasoning over time, and at some specific models for dealing with it.

The best known example is probably Kalman filtering (Section 4.4), which has received much attention for its applications to engineering, cybernetics and the military; their use for general insurance has already been explored by De Jong and Zehnwirth [JZ83]. Two other related examples are hidden Markov models (Section 4.3) and dynamic Bayesian networks (Section 4.5). We here summarise the advantages and disadvantages of the different methods:

- Kalman filters crucially depend on the assumption that both the sensor (observation) model and the transition model are multivariate Gaussian. Kalman filters are also an inherently linear model. Although the linearity assumption can be relaxed when the non-linearities are weak and therefore the evolution of the parameters over time can be approximated as a linear function locally (extended Kalman filter), and some cases of non-linearities can be addressed by switching Kalman filters, the Kalman filter model is simply not general enough to model all interesting situations in general insurance. In the practical case we looked at (reserving), many of the most relevant uncertainties over time depend on factors that are strongly non-linear and non-Gaussian, such as future changes of legislation, future changes in the reserving guidelines, sudden change in the riskiness of an account, etc. Some (although not all) of these non-linearities can be incorporated in a general probabilistic model, but it is awkward to address them with Kalman filters;

- dynamic Bayesian networks are a far more powerful method than Kalman filters:

  - all Kalman filters can be represented as dynamic Bayesian networks or as hidden Markov models, but not vice versa;
  - serious non-linearities (for example, changes of reserving guidelines, judicial decisions) can be addressed by dynamic Bayesian networks with both discrete and continuous variables;
  - numerical methods such as Markov Chain Monte Carlo can be used for approximate inference in dynamic Bayesian networks;

- hidden Markov models and dynamic Bayesian networks are equivalent formulations – however, dynamic Bayesian networks are more compact and allow gains from the sparsity of the connections between the random variables.

It should be stressed that *all* temporal models as we have presented them here have some limitations:

- in all cases, a prior model of the possible future changes is needed. This can be linear, or locally linear, or allow for strong non-linearities, but in all cases we need to have some information on the transition model. If the distribution $Pr(X_{t+1}|X_t)$ is simply unknown, there is not much we can do except rely on the latest data. This means that the past is completely irrelevant for the future. Even the most cautious actuary would not go as far as that, but there are times where the rules of the game change quickly and our ability to model future changes and the "unknown unknowns" is very small;

- the other assumption we have made is stationarity: that means that $Pr(X_{t+1}|X_t)$ does not depend on t. For much the same reasons as those listed above – the rules of the game sometimes change, new unexpected phenomena appear – this apparently harmless assumption is in certain circumstances not tenable and this can severely limit our ability to track parameters.

Chapter 5

Making decisions in an uncertain environment

> Makka Pakka, it's time to wash faces!
> *A face-washing agent in the Night Garden environment.*

Learning from data, incorporating expert knowledge, updating our knowledge in time... all these activities, which we have described in the previous section, help us understand risk. Ultimately, risk professionals need to understand risk because they have to make informed decisions. They are, in the language of computational intelligence, “intelligent agents”, because they learn from the environment and they act on it. In our case, we can speak of “risk agents” – that is, an intelligent agent dealing with risk.

Whether you want to buy a car policy, choose an investment, make a business plan for an insurance company, decide on a reinsurance purchase, you are a “risk agent” who has to make decisions in an uncertain environment: and, except in trivial cases, making decisions is not a one-off activity but a regular one: for example, an insurance company has to decide on the amount of reserves to put aside for future claims on a regular basis, perhaps on a quarterly basis, and has to decide on price changes perhaps on a monthly basis, if not almost in real time.

In this section we address the problem of a risk agent making sequential decisions in an uncertain environment. This is a well-know problem in computational intelligence: ”Designing an intelligent agent which can move in an environment making the best decisions i.e., the decisions which maximise utility”. Notice how the concept of utility has crept in here in a purely computational intelligence context, before even considering applications to the actuarial context.

The main recommended reading for this section is the book by Russell and Norvig [RN03], especially the very general Chapter 2 ("Intelligent agents") which deals with the basic notions of agents and task environments, and Chapter 17 ("Making complex decisions").

5.1 Practical (if fanciful) general insurance example: R-Age

It is helpful to have a specific example in mind while reading the following definitions and results, therefore we'll assume that our risk agent is an insurance company and that its objective is to maximise the amount of capital in the long run – that is, to become richer!

This insurance company was once called Risk Agents Ltd, but for marketing reasons it has been recently rebranded as R-AGE and pundits reckon it won't be long until the hyphenation is dropped. R-AGE is a niche company that sells a single product so its financial fate is completely determined by the success of this product. The company has a very simple rating system – it charges each customer the same fee for the policy! There is little incentive to have a better rating system as R-AGE happens to be the only player selling this product in the market, and its customer base is quite homogeneous anyway.

We will not have a separate section on R-AGE in this chapter but considerations on R-AGE will be intertwined to the main narrative, to make this otherwise abstract subject a little less dry.

5.2 Individual agents and the environment

Each risk agent is assumed to be autonomous and to incorporate strategies to interact with the environment.

The type of environment in which the agent moves is crucial. An environment can be (see [RN03, Woo01]):

- *Fully v partially observable.* A fully observable environment is simply one in which the agent always knows where it is. In the financial world we usually have partially observable environments as we don't usually know for sure what our financial position really is (this is perhaps clearer in the case of long-tail business, where it takes a long time for the financial position of a given underwriting year to become finalised).

- *Deterministic v stochastic.* A deterministic environment is one where the outcome of our actions is known for sure (in terms of the state the agent will be in). This is never the case in insurance: once for example we set the price of one policy we cannot know for sure what the volume sold and the claims to be paid will be.
- *Static v dynamic.* A static environment is simply one that does not change with time, except for the actions of the agent. A dynamic environment has other processes operating in it, and these change the environment in ways that the agent cannot control. Needless to say, the financial environment is highly dynamic. However, in some cases it may be necessary, or convenient, to assume that the environment does not change in the short term.
- *Discrete v continuous.* A discrete environment is one with a fixed, finite number of states and of actions that can be performed. Again, the type of environments we are interested in are continuous. However, this distinction is not crucial, as we can usually approximate continuous environments with discrete ones.

5.3 Markov decision processes

Although there is no such thing as a fully observable environment in general insurance, it is useful to start considering this type of environment, because it prepares the ground for dealing with partially observable environments.

In this section we therefore assume that the environment is fully observable, stochastic, static and discrete. In our example, we are basically assuming that R-Age is so good at pricing that it is able to determine the parameters of the loss distribution with infinite accuracy. It is also so market-wise that it knows the demand curve for that policy with infinite accuracy, so that it knows exactly what the volume sold will be for a given price. However, underwriters at R-Age fall short of omniscience in that they cannot predict the actual losses incurred for next year – only the distribution from which they are drawn.

These can be modelled by Markov decision processes. A *Markov decision process* is defined by:

- An *initial state* S_0
- A Markovian *transition model* $T(s, a, s')$: at each time t, an agent will be in state s and will be able to perform an action a. As a consequence, it will move to state s with probability $T(s, a, s)$. This is called a *transition model.* The model is Markovian in the sense that the probability of reaching s' from s is assumed only to depend only on s and not on the states that preceded s.
- A *reward function* $R(s)$, establishing the reward received by the agent in state s.

How is this relevant to R-Age? The state s in which R-Age is at time t may be simply defined as a pair $s = (C_t, V_t)$ where C_t is the capital available and V_t is the volume sold at time t. Time is discretised and is in years. The initial state is the initial capital C_0 and the initial volume V_0. Actions are simply decisions on prices: therefore a is another symbol for P_t, the price charged at time t. The transition probability $T(s, a, s')$ is the probability that if the price is set to a the capital will become C_{t+1} (the volume V_{t+1} is deterministic in this simplified setting), and will depend exclusively on the aggregate losses S_{t+1} incurred between t and $t + 1$ (for simplicity, we assume that these include expenses):

$$C_{t+1} = C_t + V_t P_t - S_t \tag{5.1}$$

If we know the distribution of S_t – for example, we might know that it is a lognormal distribution – we automatically know the distribution for C_{t+1} given C_t and P_t: that is, the transition probability $T(s, a, s')$.

Note how this is beginning to look like ruin theory! The main difference for now is that premium changes and so does volume.

Equation 5.1 also allows us to define the reward for each state as:

$$R_t = V_t P_t - S_t \tag{5.2}$$

which is simply the profit during year t.

Now that we have defined what a Markov decision process is, we need to measure how good a decision is, so as to choose the best one. Or rather (since we are not interested in one-off decisions but in sequences of decisions), what we need is a method to decide how good a *policy* (that is, a prescription that specifies what the agent should do in any state it might reach) is. Since the word "policy" is already ubiquitous enough in this paper for other reasons, we will call this a *strategy* rather than a policy, and will indicate it as σ. $\sigma(s)$ is the action prescribed for the agent when this is in state s.

In order to decide what the best strategy is, or simply to say whether a strategy is better than another, we need an ordering on the set of strategies – in other terms, a *utility function.* Given the stochastic nature of the environment, the quality of a strategy σ is measured by its *expected* utility, where the expectation is calculated over all possible environment histories generated by σ.

Quite naturally, an *optimal strategy* is simply one that achieves the highest expected utility. Interestingly, if the utility function is constructed wisely, strategies which yield high rewards in the short-term but are too risky will be penalised. As a result, there is a familiar balance to be struck between risk and reward, and this arises simply as a consequence of the stochastic nature of the problem. As Russell and Norvig note in [RN03], "the careful balancing of risk and reward is a characteristic of MDPs that does not arise in deterministic search problems; moreover, it is a characteristic of many real-world decision problems.".

How can we assign a utility function $U_h([s_0, s_1, \ldots s_n])$ to environment histories $[s_0, s_1, \ldots s_n]$? This depends, first of all, on whether the horizon for decision-making is finite (only rewards accumulated up to year N are relevant) or infinite (the whole future history is relevant). Finite horizon problems are more complicated because they are inherently non-stationary: the optimal decision will depend on how close you are to the deadline.

We will assume here that the horizon is infinite – which simply means in our case that you don't know *a priori* how long you're going to stay in business, not that you'll be in business forever.

We will also assume that the agent's preferences between state sequences are stationary, meaning that if the utility of sequence $[s_1, s_2 \ldots]$ is greater than that of $[s'_1, s'_2 \ldots]$, then the utility of $[s_0, s_1, s_2 \ldots]$ (identical to the former sequence except for the addition of the initial state s_0) is greater than that of $[s_0, s'_1, s'_2 \ldots]$ (which starts with the same state s_0).

It can be shown that under these conditions there are only two ways in which we can assign utilities to sequences:

- *Additive rewards.* The utility of a state sequence is

$$U_h([s_0, s_1, s_2 \ldots]) = \sum_{j=0}^{\infty} R(s_j) \tag{5.3}$$

- *Discounted rewards.* The utility is defined as

$$U_h([s_0, s_1, s_2 \ldots]) = \sum_{j=0}^{\infty} \gamma^j R(s_j) \tag{5.4}$$

  where γ, the discount factor, is a number between 0 and 1.

Although both Equations 5.3 and 5.4 are theoretically correct, the former choice is not acceptable in practice because it leads in general to infinite utilities (a solution would be of course to define utilities in terms of averages rather than of sums). Therefore, it looks like there is only one sensible way to define utility, and this is Equation 5.4.

This is a somewhat surprising result: without ever mentioning the time value of money and without any other actuarial consideration, but simply assuming an infinite horizon and the stationarity of the agent's preference over state sequences, we have found that **the only sensible way in which we can define utility is as the sum of the present value of future rewards**!

We are now in a position to define the optimal strategy σ^* as the one which maximises the expected utility as defined in Equation 5.4:

$$\sigma^* = \text{argmax}_\sigma E(\sum_{t=0}^{\infty} \gamma^t R(s_t) | \sigma) \tag{5.5}$$

It is obvious how this applies to R-Age: the utility is simply the present value of future profits. As to what discount factor γ to use, that is quite a usual problem – in a normal situation, one solution is using the return expected by investors for the level of riskiness of this company (in this case, this policy). In the specific case addressed in this section – for which the only real uncertainty is in the randomness of the aggregate losses, whose parameters are assumed to be known with certainty – it would probably be more appropriate to choose a lower discount rate, such as the risk-free discount rate plus a "premium" related to, say, the coefficient of variation of the aggregate loss distribution.

The solution to Equation 5.5 can be found with the so-called value iteration algorithm to solve Bellman's equations, which are recursive equations describing the utility of a state sequence. The value iteration algorithm is guaranteed to converge to a unique solution, and is efficient (see [RN03], Section 17.2). Although theoretically interesting, this is of little relevance to us, as Markov decision processes are merely a step towards partially observable MDPs, which require more advanced tools for calculating the optimal strategy.

5.4 Partially observable MDPs

In practice, R-Age will not have be able to determine its state (C_t, V_t) with certainty for $t > 0$: for example, when writing long-tail business some claims will only be known or settled many years after their occurrence. The estimate of the current capital depends on the reserve estimates, which are uncertain (owing to IBNR, IBNER, etc). From an accounting point of view there is a capital declared at the end of each year, but the real financial position of the company is not known with certainty.

In the language of intelligent agents, what we are saying is that the environment is not fully observable and that the agent does not know for certain which state it is in. This is modelled by the so-called partially observable MDPs, or POMDPs – which is usually pronounced "pom-dee-pees", so as to make them eligible characters of "In the night garden" [InT].

By way of example, [RN03] specifies an agent which has no sensors at all and still has to make the optimal decision. This may seem far-fetched but much London market business has to be priced in similar conditions: and the story of asbestos-related claims shows quite clearly how concrete risk agents (in this case, the insurance companies) have often to make decisions while they're in the dark with respect to how much they will be asked to pay in the future.

A POMDPs requires the following ingredients:

- A transition model $T(s, a, s')$
- A reward function $R(s)$

- An observation model $O(s, o)$ that specifies the probability of perceiving the observation o in state s
- A belief state, i.e. a mapping between actual states s and the probability $b(s)$ assigned to each of them. Note that the belief state can also be defined as *the posterior distribution over the current state, given all evidence to date*: or, using the notation of our section on temporal models, $Pr(X_t|e_{1:t})$.

As new observations are made, the belief state is updated, according to this equation, which is simply an application of Bayes' theorem, with a non-standard format.

$$b'(s') = \alpha O(s', o) \sum_s T(s, a, s')b(s) \tag{5.6}$$

which can be abbreviated as

$$b' = \textsc{Forward}(b, a, o) \tag{5.7}$$

What makes POMDPs work is that the optimal strategy σ^* depends only on the agent's current belief state and not on the state it is in (which is not known anyway). This is how a POMDP makes decisions:

1. Given the current belief state b, execute $\sigma^*(b)$ (the optimal strategy given the belief state)
2. Receive observation o
3. Set the current belief state to $\textsc{Forward}(b, a, o)$ and repeat

Interestingly, the problem of solving a POMDP on a physical state space can be reduced to that of solving an MDP on the corresponding belief state space:

- The probability of reaching b' from b given action a is

$$\begin{aligned}
\tau(b, a, b') &= Pr(b'|a, b) = \\
&= \sum_o Pr(b'|o, a, b)Pr(o|a, b) = \\
&= \sum_o Pr(b'|o, a, b) \sum_{s'} O(s', o) \sum_s T(s, a, s')b(s)
\end{aligned}$$

where $Pr(b'|o, a, b)$ is 1 if $b' = \text{FORWARDS}(b, a, o)$, 0 otherwise.

- The reward for belief state b can be defined as:

$$\rho(b) = \sum_{s} b(s)R(s) \tag{5.8}$$

(that is, the reward for a belief state is the expected value of the reward for individual states)

These equations have the same form as the definition for fully observable MDPs as stated in the previous section: the only difference is that the state s is replaced by the *belief states* $b(s)$. As promised, we have reduced the problem of solving POMDPs in the state space to the problem of solving MDPs in the belief space. It can also be shown that an optimal strategy $\sigma^*(b)$ for this MDP is an optimal strategy for the original POMDP.

The difficulty, however, is that the space of belief states is continuous and usually high-dimensional. Solutions for this case have been developed as well, but the computational complexity of these solutions becomes rapidly prohibitive as we increase the number of states: the problem falls in the class of PSPACE-hard problems. The definition of the complexity class PSPACE and of PSPACE-hard problems is quite technical and beyond the scope of this dissertation: suffice it to say that PSPACE-hard problems are at least as hard as NP-complete problems and are widely thought to require exponential time to solve (see [Pap94]).

Luckily, approximate methods for solving POMDPs have been developed, making use of the so-called "'dynamic decision networks"' (DDNs), which we will describe in the next section.

Before going into that, however, let us see how POMDPs apply to R-AGE. As we said at the start of this section, the company will not know its financial position for each year with certainty, mainly because of reserve uncertainty. We can certainly assume that the financial position (C_0, V_0) will be known, but any successive year will be uncertain. How does the company decide what to do in order to maximise capital in the long run?

The belief state b in any year different from 0 will be determined among the other things by the reserve uncertainty.

Instead of having a state (C_t, V_t), in the partially observable case we have a belief state $b(s)$ which gives the probability of each state and that we can write for R-AGE as $Pr(C_t)$. The reward for belief state $b(s) = Pr(C_t)$ is defined as (in the case when states can be discretised)

$$\rho(Pr(C_t)) = \sum_{C_t} Pr(C_t)(V_t P_t - S_t) \tag{5.9}$$

To have a fuller picture, we should actually include in the description of the state the parameters of the aggregate loss distribution, which are also not known with certainty: $b(s) = Pr(C_t, \mu, \sigma)$ (λ and μ are for example the parameters of a lognormal distribution modelling the total losses) or $b(s) = Pr(C_t, \lambda_t, \sigma_t)$ if we want to allow for the possibility that parameters actually change in time.

As for the general case, for R-AGE the optimal strategy does not depend on the current state but on what we know about the current state. The three-steps POMDP decision process can be rewritten for R-AGE as:

1. Given the current estimated distribution $Pr(C_t, \lambda_t, \sigma_t)$ for the capital C_t and for the estimated parameters of the total loss distribution, set the optimal price P_t^*
2. Receive observations on reported losses, paid losses, volumes sold, etc.
3. Revise the reserve estimates, the total loss parameters, etc thus producing a new estimate $P'(C_{t+1}, \lambda_{t+1}, \sigma_{t+1})$

5.5 Dynamic decision networks

As mentioned in the previous section, POMDPs are difficult to solve. However, approximate methods are available that provide sub-optimal solutions. Specifically, *dynamic decision networks* provide a comprehensive approach to agent design for partially observable stochastic environments.

The approach can be outlined as follows:

- The agent is represented as a **dynamic decision network (DDN)**, that is a dynamic Bayesian network (Section 3.3) augmented with decision and utility nodes. Decision nodes are nodes where decision makers have a choice of action – e.g., what price R-AGE should charge for its policies. Utility nodes are related to the utility of the state (including the rewards received);

- A monitoring algorithm similar to that described in Equation 4.2 is used to incorporate new evidence, the effect of actions (which is treated as evidence) and to update the belief state, e.g. (in the case of R-AGE) the probability distribution of the capital held;
- Decisions are made by considering possible courses of action and then choosing the one that leads to the best results (in terms of the utility function).

Summing up the notation:

- X_t is the state variable (a vector, in general) and E_t is the evidence variable, as usual. In the case of R-AGE, $X_t = (C_t, \lambda_t, \sigma_t)$ and E_t is the collection of reported claims, paid claims, information on volume sold, etc.
- A_t is the action at time t, e.g. setting the price P_t for the policies
- R_t is the reward at time t, e.g. the profits in year t
- U_t is the utility of the state at time t, i.e. (by definition) the sum of discounted future rewards.

The transition model for this network is $Pr(X_{t+1}|X_t, A_t)$ ($T(s, a, s')$ in the language of POMDPs) and the observation model is $Pr(E_t|X_t)$ ($O(s, o)$ in the language of POMDPs).

Figure 5.1 provides the generic structure for a DDN, and its intepretation for our R-AGE example. The network has been projected for three future periods to visualise the way in which a network like this is "solved": since the problem of finding the optimal strategy is intractable, the way one produces a (sub-optimal) strategy is by a look-ahead method:

1. Assume we are at time t
2. Set $k = 1$
3. Explore all possible actions at time $t + k$
4. For a given action, the updated *belief state* (not the actual state!) is uniquely determined by the monitoring algorithm explained above
5. Explore all possible "environment actions", i.e. observations

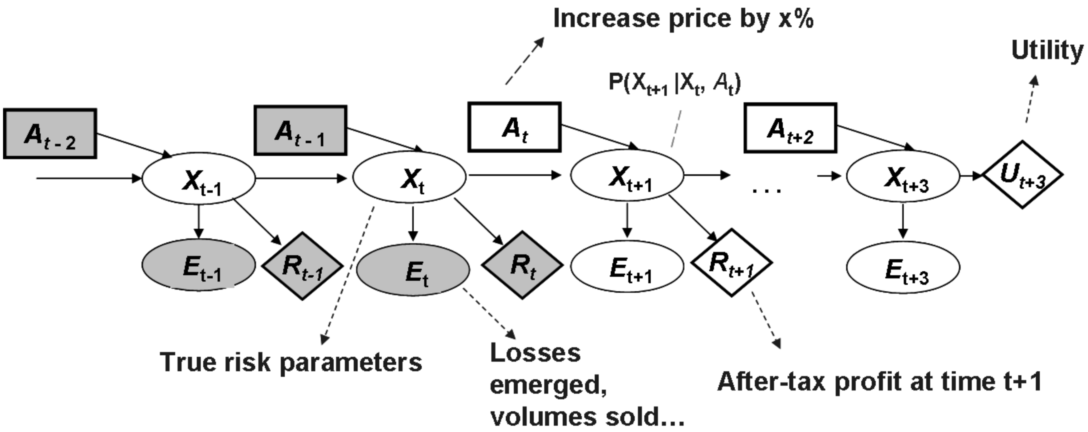


Figure 5.1: An insurance company pricing decision process seen as a dynamic decision network.

6. Calculate reward
7. Set $k \leftarrow k+1$ and go back to Step 2, until desired look-ahead depth is achieved
8. When desired look-ahead depth is achieved, calculate the utility function in approximate form

Note that we are using an approximate utility in the method above: if we knew the exact utility, we wouldn't need a look-ahead algorithm! We could directly calculate U_t. Also note that looking ahead only a few steps is usually sufficient when the discount factor used is small enough, as the rewards beyond a certain level might be negligible.

Coming back to our R-Age example, please note that:

- This exhaustive search is of course possible only for a discrete number of states. This is not a major difficulty for our example as we can only put reasonable upper and lower bounds to prices and losses and discretise the possible outcomes. Note that the computational complexity of exhaustive search is $O(|D|^d|E|^d)$ where $|D|$ is the number of available actions, $|E|$ is the number of possible observations, and d is the depth of the look-ahead search, therefore one needs to be sensible with the discretisation.
- Rather than an exhaustive search we might sometimes resort to a stochastic search where only certain actions and observations are explored, drawn at random – this is the strategy usually applied in dynamic financial analysis exercises.

- Rather than considering all possible actions, one can – if actions/observations are drawn at random – consider only certain possible strategies, such as “increase the price by this amount if this happens” and test them.
- Note that although we haven’t focused here on this aspect, the concept of “ruin” can be easily put into the picture by giving “ruin” a highly negative reward.

Overall, POMDPs and especially dynamic decision networks represent a good way of representing risk agents such as an insurance company, at least in situations that are not too complex. Various approximation methods – beyond those illustrated here – can be used to make near-optimal decisions with reasonable computational complexity. The interested reader is referred as usual to [RN03] for a more thorough treatment of these techniques (see especially Chapter 21 on reinforcement learning).

The reader will have noticed analogies with ruin theory, dynamic financial analysis and pricing optimisation. As a matter of fact, all these theories/activities can be put in the context of risk agents making decisions in an uncertain environment:

- ruin theory – at least basic ruin theory – is a basic ingredients of POMDPs dealing with risk. POMDPs extend ruin theory in the sense that they also enable us to consider the effect of decisions and other uncertainties (including model and data uncertainty) in a Bayesian context;
- DFA can be put into the context of POMDPs. DFA is a special case of POMDPs where utility can be calculated by stochastic simulation and the effect of possible actions and observations are considered – possibly interactively;
- pricing optimisation can be put into this context as well: in this case the demand modelling is given special emphasis.

Chapter 6

Modelling collective behaviour

> The real world is muddy and messy and full of things that we don't understand.
> Freeman Dyson, *Heretical Thoughts about Science and Society (2007)*

We now get to the most difficult aspect of general insurance of all – the closest to the real world – that of risk agents interacting in a market. In the previous section, we have tried to model risk agents acting in an environment: however, the environment was considered as an amorphous "blur" which could somehow be modelled as a whole. This is for example the idea behind demand modelling.

To a certain extent, this is inevitable, especially when personal lines business is concerned: one can hardly imagine to model millions of customers as individual entities. However, not all aspects of the insurance market can be explained like this and when the number of players is limited the granularity of the environment cannot be ignored. The need for collective behaviour analysis is clear once we observe that certain effects, such as the underwriting cycle and asset bubbles, are genuinely collective. There is no explanation of either phenomena at the individual level.

The essential point is that a player in the insurance market simply cannot be successful if its strategy ignores the interaction with the other players. Pricing is an obvious example: it is not just a matter of understanding the aggregate loss distribution, or the expenses, or optimising the price by using demand modelling. The ultimate decision on how to price insurance cannot ignore how the other insurers are pricing the same or similar products - it is a strategic decision which is informed by many things, among which the technical price. And this is only truer when the number of players is limited and the number of clients is smaller as well.

For example, the risk agent/environment paradigm does not really explain the existence of an underwriting cycle, nor has the ability to model the reinsurance market, where a very limited number of players is involved who cannot be averaged in a demand curve.

What techniques can we use to analyse collective behaviour? A few possibilities come to mind:

- methods borrowed from statistical or many-body physics, biology, etc. Physical systems exhibit genuinely collective behaviours such as coherence (which will remind many of ”systemic” effects), phase transitions, etc.;
- game theory, especially where it concerns the interaction of many players, such as in auctions;
- multi-agent systems (MAS): an important emerging paradigm in AI, MAS are collections of individual agents with specified behavioural rules;
- genetic algorithms, which adopt the evolutionary paradigm of the survival of the fittest as a problem-solving paradigm. These can actually be considered as a special case of multi-agent systems.

Even leaving aside the obvious argument that people tend to be less predictable and more devious than molecules, in statistical mechanics we usually deal with an extremely large number of identical objects. This is rarely the case in general insurance, where - even in the case of personal insurance - there might be millions of customers with nearly identical purchasing behaviours but only a few hundreds insurers with very different strategies. In reinsurance, we have hundreds of insurance companies buying from a few dozens of reinsurers. Despite these caveats, certain concepts - such as phase transitions and coherence - may still translate to the financial domain, and be anyway an inspiration in the back of one’s head although the analogy will not be perfect.

As to genetic algorithms, they have been used for modelling purposes and for problem solving before. However, genetic algorithms should be seen more as an optimisation technique rather than as a tool for understanding collective behaviour. As neural networks, they suffer from the problem of "opacity": they may produce good results but not necessarily shed light on the rationale behind the decisions. A recent application of genetic algorithms to capital modelling as an optimisation tool was recently presented at GIRO 2009 by Haslip and Kerley [HK09].

MAS are - according to the author - the most promising technique for modelling collective behaviour, as they allow for any level of complexity of the individual players, and they allow to model the interaction between agents.

- Making decisions by an intelligent agent in an uncertain environment can be seen as a Markov decision process. If a reward function (e.g. profit) is defined for every state the intelligent agent is in, and the utility of the state is taken to be the discounted sum of rewards, the optimal behaviour of an agent can be reduced to that of maximising expected utility;
- when many different agents are involved, game theory provides a framework to model the interaction between agents. Solutions of the collective games are Nash equilibria - strategies where no agent has an incentive of deviating from the specified strategy.

This chapter continues the exploration of Chapter 5, where we investigated the decision-making process for an individual agent in an uncertain environment, and is organised as follows.

In Section 6.1 we consider the case where this uncertainty is at least partly due to the simultaneous interaction between different agents, and we see how stochastic simulation can be used for both agent design and mechanism design. Section 6.2 illustrates the role played by game theory. Practical examples are discussed in Sections 6.3, 6.4, and 6.5.

As for background reading, one can do no better than read the classical book by Russell and Norvig [RN03], which is (fairly) advertised on its cover as *The Intelligent Agent Book*: it is actually a book on all aspects of artificial intelligence but the concept of intelligent agents informs all parts of a book, and a full section is devoted to the aspects of intelligent agents that are of more direct interest to us (Chapter 17). A book fully devoted to multi agent systems is [Woo01]. This contains a good section on game theory, which receives a shorter treatment in [RN03]. However, this book is mainly aimed at software engineers and is less concerned with the quantitative aspects of intelligent agents.

6.1 Multi-agent systems

Multi-agent systems are simply collections of intelligent agents – as defined for example in Section 5. Multi-agent systems can be used in a cooperative or non-cooperative fashion. For example, agents may cooperate to solve a given problem, by breaking it up into smaller tasks: this is a familiar use of agents in a software engineering context. Alternatively, they can be made to compete with each other. In this latter non-cooperative framework, which is the most interesting to us, there are two main ways in which we can use multi-agent systems:

- Agent design: designing agents that deal with other agents with an optimal strategy, so as to maximise their utility.
- Mechanism design: designing rules to regulate the interaction between different agents in order to maximise some overall measure of utility.

If we look at insurance companies as agents, agent design is what is required for an insurance company which wants to thrive among its competitors. The example of R-Age in Section 5 is a case in point. Another example is that of an individual wanting to buy a household policy. The example can even be more extreme: an actual software agent making the best purchase of insurance in the internet. This example is not far-fetched because this type of agents has actually been designed for electronic shopping, although perhaps not for the insurance market.

Mechanism design is instead the job of the regulator. When devising new regulations, it is important to understand what the effect of the regulation is going to be, and to test different types of regulations so as to reduce the instability of the system. This effect can only be understood at the collective, and not at the individual, level.

Some authors [ZO04] distinguish three different scales at which agents will exhibit different behaviours. This idea originates in the field of software engineering but still provides good food for thought.

- Macro scale, at which we have a large number of agents with similar or identical operations
- Micro scale, at which we have a small number of agents with customised operations

- Meso scale, which is not really an intermediate scale but rather a scale that requires considering aspects of both micro and macro scale. Typically, that is the relevant scale when we design agents at the micro-scale, with customised behaviour, that will then have to interact with a large network of agents (macro scale).

In the general insurance market, we are almost never at the pure macro scale: even in the case of personal insurance, where there are millions of customers with roughly the same behaviour, the interaction is with a limited number (in the order of hundreds) of insurance companies, which will have individual behaviours and strategies. According to the classification above, this would be a meso scale.

In some cases, such as that of reinsurance, the most appropriate scale would probably be the micro scale, as each agent pursues different strategies, has different technical capabilities, size, etc.

In all cases, the interaction with other players in the market is crucial. This interaction requires usually both one-to-one interactions or in some cases simply collecting intelligence on what other players are doing.

As to the types of agents one needs to consider, the general insurance market hosts several of them, each with an essential role to play:

- retail customers – what most of us are
- commercial clients
- insurance companies/Lloyd's syndicates/etc
- reinsurance companies
- insurance brokers
- reinsurance brokers
- the financial regulator
- banks
- investment funds
- ...

An example which follows this spirit although it ignores the formalism of multi-agent systems is the paper presented at GIRO 2008 by Greg Taylor [Tay08], which investigates the effect of catastrophes and of regulation on producing cyclical effects such as the underwriting cycle.

6.2 Game theory

When analysing collective behaviour, one simply cannot ignore game theory. Game theory plays a role whenever there are agents that have to adopt successful interaction strategies, and can be used for both agent design and mechanism design.

Basic concepts such as a payoff matrix, Nash equilibrium, dominant strategies, Pareto optimality, zero-sum games should be familiar to most actuaries and they will not be explained here, also because they are of little direct application in this section. However, the term 'game theory' is an umbrella term for many different ways of analysing interaction. The “type” of game theory which is perhaps the most interesting for modelling the collective behaviour of insurance market is that which involves:

- repeated games, in which agents meet one another repeatedly and keep memory of the previous encounters;
- several different agents (possibly a large number) making decisions simultaneously;
- information which is only partially observable, in the sense that each agent has only partial information on what its opponents are doing.

Games in this type of setting are called **games of partial information**. Popular examples include card games such as poker and bridge, in which each player can only see a subset of the cards, or the modelling of nuclear wars, where the location of the opponents' weapons is unknown. Games of partial information can be solved theoretically by similar techniques to those used in Section 5.5, where a solution maximising utility was obtained by a look-ahead technique. A key difference is that one's belief state is observable whereas that of the opponents are not.

Some practical algorithms have recently been devised for playing toy versions of poker. These involve bluffing and randomised strategies. Randomisation plays a key part in one's strategy as it prevents the opponents to gather too much information on one's strategy based on one's actions.

According to Russell and Norvig, there are two main reasons why game theory has not been widely used for agent design:

- In a Nash equilibrium solution, a player is assuming that the opponents will play the equilibrium strategy, without being able to incorporate any prior belief that the player might have of what the opponent(s) might do. To address this problem, the notion of Bayes-Nash equilibrium has been introduced which explicitly incorporates our prior beliefs about the other players' likely strategies.
- No good way is currently known to combine game theory with POMDP theory in the framework of agent design.

For these reasons, game theory has been more successful in mechanism design, by which one attempts to maximise some global utility, basically attempting to discourage behaviours on the part of individual agents which might decrease the overall utility. This can be obtained by the imposition of rules that incorporate all externalities – that is, effects on global utility not recognised in the interaction between agents – explicitly. This has been successfully used for example in designing auctions [Woo01, RN03]. Another example, which is more directly related to general insurance, is briefly described in Section 6.5, and relates to the problem of determining the optimal allocation of quotas of risk among insurers that evaluate risk differently.

In most cases concerning general insurance, an analytical solution of games involving risk agents will of course be impossible, given the large uncertainties and the large number of variables involved. In the author's opinion, the most promising application of game theory to multi-agent systems in insurance is through the stochastic modelling of agents, much along the lines of the world done by Axelrod [Axe84] on the evolution of cooperation.

In a much cited and celebrated example, Axelrod devised a tournament among agents (represented as programming code) that would meet repeatedly at random and could choose at each encounter whether to "cooperate" or "defect". The payoff matrix was devised in such a way that mutual cooperation was preferred to mutual defection, whereas the best outcome for an agent would occur when the agent defected while the opponent cooperated. The idea was to have a repeated version of the famous prisoner's dilemma, in which the most rational decision is for both players to defect. The fact that the change is repeated creates a long-term incentive to cooperate (to avoid retaliation), which makes the game more realistic.

Since there are infinitely many strategies that would have had to be tested, researchers were invited to submit their own programs and see how they would perform. The winner was a simple strategy, called tit-for-tat, that always starts cooperating and then mirrors the opponent's last move – defecting if in the previous encounter the opponent had defected, and cooperating otherwise. Tit-for-tat has proved very resilient in time as new tournaments were devised, and has proved the point that a good strategy should be simple, should prefer cooperation and should punish defection.

The key idea of the Axelrod experiment that makes it interesting for us is its stochastic nature (agents meet at random) and the idea that rather than trying to design an optimal agent based on theoretical considerations (which proves to be extremely complicated even in simple settings such as this – let alone when modelling insurance companies), one should design "tournaments" to test different strategies, and tentatively adopt the strategy which performs better in a number of tournaments. The strategies might be based on what we believe our opponents might do.

A possible application of this to general insurance is where there are two types of agents, customers and insurers, and customers may submit claims, which may be either genuine or fraudulent, and insurers have the choice of paying or not paying, whether they believe they are fraudulent or not. A tournament involving random repeated encounters of customers and insurers may be designed and may give insight on what an insurance company should do for a given perceived likelihood that the claim is fraudulent. Several variations can be considered, depending for example on whether the knowledge of one company rejecting a claim passes on to other customers (and with what probability), and on the degree of communication between insurers (which might share a common database of suspect fraudulent claims).

Another application, which abandons the idea of one-to-one encounters, is that of n insurers selling policies to m customers and competing for market share. The insurers will try to set the price that will maximise their profit. The trade-off is of course between volume of policies sold (which can be increased by decreasing the unit price of the policy) and the profit per policy (which can be increased by increasing the unit price of the policy). Decisions are sequential as in 5, and for simplicity can be taken at discrete times, e.g. months, based on both the past statistics on the policy and the prices set by the competitors. To model this situation, it is not necessary to consider random one-to-one encounters, whether between insurers or between an insurer and a customer, but the idea of a tournament of strategies can still be adopted. This will be explored in more depth in Sections 6.3 and 6.4.

Notice how in this latest setting only a few core ideas of game theory are actually retained while most of the concepts of payoff matrix, Nash equilibrium etc are not applicable.

6.3 Practical general insurance example: The market for personal insurance

Consider the problem of determining the pricing strategy for a personal lines insurer – we can use R-Age again. For simplicity, assume as in Section 5.1 that R-Age is only engaged in one line of business. To make things even simpler, we actually assume that the insurer only sells a single simple product, e.g. household buildings insurance and that there is no product or policyholder differentiation. We also assume that cover is compulsory, as motor insurance is in the UK, so that every customer must buy his/her policy from exactly one of the insurance companies in the market.

The market is also simplified to its core, with only insurers (R-Age and competitors) and customers trading in it. The environment is obviously stochastic (losses come from a stochastic process), works at the macro/meso scale with a disproportionate number of customer-type agents with respect to insurer-type agents.

Here is a possible simple model for the personal lines insurance market:

- Each insurer is modelled as a POMDP, much in the same way as in Chapter 5. The main difference is that each insurer also has some knowledge of what the other insurers do. The number of insurers is not necessarily constant: when the average loss ratio is small, it is likely that some insurers will join the market, and some insurers may exit the market either because they're not satisfied with their level of profit or because their capital has gone below a certain regulatory threhold. We assume that insurers vary as to their ability of calculating the technical price, and that this ability depends only on the volume of policies sold – this is a proxy for many things, one of which is parameter uncertainty (which reduces with the size of the portfolio), but also of the fact that larger insurers are usually more sophisticated because they invest more heavily in technical personnel and in systems. We can assume that the business is short tail and that all losses are reported and settled in the same year in which they occur.

- Each customer can also be modelled as an agent. In its simplest form, the customer's choice would be fully price-driven, and there will be some degree of stickiness – so that the customer will not constantly jump from one insurer to the other in his/her quest for the best price; rather, he/she will jump with a likelihood which increases with the price differential. We can assume that the customer has full knowledge of the prices in the market. This is not usually the case but aggregators are getting us there. Since the price is unique given an insurer, we are not interested in the customer's losses. The utility of the customers is simply the discounted sum of the premiums paid, after a change of sign.

- The structure of the market can be represented as a bipartite graph, that is a triple $G = (U, V, E)$ where $U = (u_1, \ldots u_m)$ and $V = (v_1, \ldots v_n)$ are two sets of vertices representing the insurance companies and the customers respectively, and E is a subset of $U \times V$ – that is, a set of edges which connect insurance companies and customers. The rule is that an insurance company u_i can be connected to many customers in V but a customer v_j can only be connected to one company. This can be represented as an $m \times n$ matrix δ whose elements $\delta_{i,j}$ are equal to 1 if v_j is a customer of u_i, 0 otherwise. A representation of this graph is shown in Figure 6.1. The matrix is such that the sum across each column is 1 and the sum across each row is equal to the number of customers of each insurance company. This matrix is useful to visualise what is happening but is quite cumbersome to deal with, especially because the graph is relatively sparse (the matrix has $n \times m$ elements but only m elements that are different from 0 – therefore a simpler represenation is that which just gives the list of customers with the corresponding company.

- The interaction between customers and agents can be modelled as a non-cooperative game with stochastic payoffs (see [FM05]).

- In this simplified setting, the regulator does not appear explicitly in the environment as an agent – so there's no direct interaction between insurers and the regulator. However, the regulator has a role in the context as the rule setter – for example, in deciding what is the level of capital below which an insurer is asked to quit the game, or in deciding what is the minimum level of capital for entering the game.

- The utility function of each insurer will be the same, and be equal to the expected sum of all future rewards – i.e., all insurers will try to maximise expected long-term capital.
- Each insurer will be able to decide what to do based not only on its losses and volume of sales, but also on the price set by its peers the year before, which is assumed to be publicly available (in this simplified setting). This provides the basis for introducing game theory – however, notice that we do not need one-to-one encounters as in Axelrod's experiment, but rather we have agents deciding what to do based on information collected on its peers. The one-to-one encounters are (implicitly) those between insurers and customers.

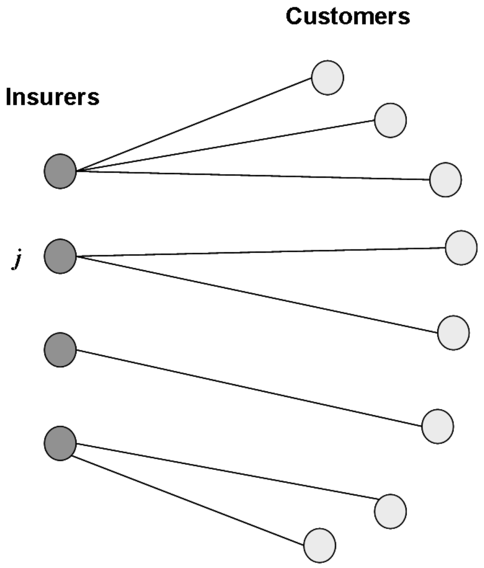


Figure 6.1: A sketchy representation of the market for personal insurance

A close-up representation of the personal lines insurance market – which focuses on the carriers – is shown in Figure 6.3.

As shown for example in [Tay08], even in very simplified settings one notices the emergence of collective phenomena such as the the undewriting cycle. Notice that the underwriting cycle probably emerges quite natural as a quasi-periodic behaviour which is typical of situations where there examples of interacting populations (think of preys and predators in biology).

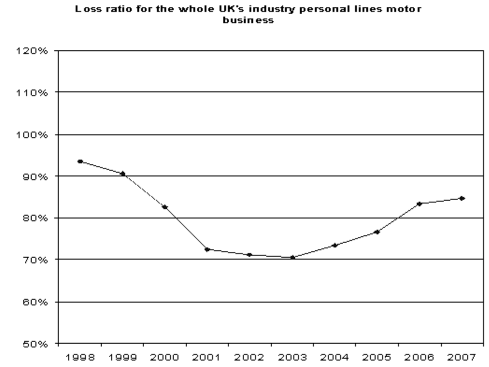


Figure 6.2: The underwriting cycle for motor. Source: FSA returns

The model above can be refined to make it more and more realistic. Here are a few ingredients that will be necessary to consider in a more realistic model:

- The inclusion of long-tail features – that is, the fact that the financial position of a company is known only after a number of years, after claims have fully developed to their ultimate value. This will make it more difficult for companies to calculate an appropriate technical price.
- The inclusion of differentiated pricing for different policyholders based on their risk profile. If we include this feature, it is important to consider the fact that the profile of each customer will be judged in slightly a different way by different insurance companies. This basically amounts to some insurers being able of more predictive accuracy than others, and so it has been partially taken into account in the simple scenario above. However, the main difference here is that customers will tend to flock to the insurers that consider them to be a better risk (adverse selection).
- The inclusion of more agents, such as intermediaries (brokers), reinsurers, reinsurance brokers, the regulator (explicitly), etc. At the very least, reinsurance should be part of the picture.
- The inclusion of a model for expenses and commission
- The inclusion of investment income in the picture, with a differentiation between different types of investment

- Limitation of the number of customers accepted by the insurer depending on the capital held

- The inclusion of more than one line of business, and of more than one territory. This will allow diversification to be taken into account, and will also allow to consider the effect of cross-subsidies, which plays a key role in competitive strategies.

- Product may not be compulsory

- The level of services offered by the company may be modelled as well, for example as a simple function of expenses in excess of claims-related expenses.

- Constraints on pricing given the capital position.

- Customer agents may incorporate more sophisticated behaviours rather than being just price-driven and lazy (stickiness). These behaviours may include taking into account the level of service offered by the insurance company, the brand popularity, etc.

The model outlined above represents a framework into which to analyse collective behaviour. Is it a practical solution? One must of course be aware that the level of complexity of even the most basic model is so high that there is a danger that we are able to reproduce well-known phenomena (underwriting cycle, etc) in a spurious fashion, simply because the number of parameters is sufficient to imitate those phenomena by appropriate tuning. At the same time, even the most complicated version of the model will pale in comparison with the messy reality of the insurance market and will not be realistic. So, why bother? There are at least two reasons to attempt this investigation anyway:

- One reason is that although stochastic simulation of market behaviour may not (actually, will never) reproduce the reality of the insurance market in an exhaustive way, it may bring to light unexpected collective phenomena;

- if we are able to reproduce known phenomena (bubbles, cycles, etc) with a very small number of parameters, and with a broad range of parameter values (that is, robustly), we may indeed have found something which has predictive value and that could be used for further investigation.

The overriding consideration should be to consider these models with much caution, as useful tools for exploratory analysis rather than rigorous means of understanding collective behaviour.

6.4 Practical general insurance example: The reinsurance market

The main difference between the personal lines insurance market and the reinsurance market are:

- **Scale**. The reinsurance market is at a different scale than the personal insurance market: there are only a few dozens reinsurers from which a given reinsurance product (e.g. excess-of-loss motor reinsurance) can be purchased, and only a few hundreds insurance companies that may be interested to buy it.
- **The role of brokers**. The role of brokers is more difficult to ignore even at a simplified level, as certain markets (such as the Lloyd's of London) are accessible only through the brokers.
- **The slip system**. An insurer may be reinsured by a number of reinsurers, each of one takes a share of the risk (e.g., the slip system in the London market). If brokers are included, there will in most cases be only one broker for each insurer, but the insurer may reinsure part of the risk directly through a reinsurer.

A representation of the reinsurance market can again be sketched using the formalism of graphs. In the simplified case where we do not include reinsurance brokers in the picture, the representation is as in Figure 6.4.

For illustration purposes, we henceforth focus on the cases which does not include brokers. A possible, simple model for the reinsurance market is as follows. We assume we are dealing with a single product (e.g. motor reinsurance), a single type of contract (e.g. one-year excess of loss, with unlimited liability, so that the only variation is the attachment point of the contract). We also assume that the market is a subscription market and works according to the slip system as the London market – every reinsurer underwrites a line (a percentage) of the risk until 100% of the risk is covered. The price is decided by the leader (no "vertical placing").

- Each insurer is modelled as an agent much in the same way as customers in personal lines insurance were modelled in Section 6.3. As for the personal lines insurance customer, the insurer here will have stochastic losses. However, the decision-making mechanism of the agent is more complicated, as it will base its decision on:

  - the overall price;
  - the composition of the panel of reinsurers in terms of security (a much-used proxy for that is the rating assigned to the company by the rating agencies), and the spread of risk between different reinsurers. This might take the shape of constraints, for example: do not use markets with rating less than $A-$, and do not give any reinsurer more than 30% of the risk;
  - how much of the risk it would like to retain (attachment point).

- Each reinsurer is modelled as a POMDP. That the environment is only partially observable is perhaps clearer here than in the case of personal lines insurance, as the uncertainty on the current financial position of the reinsurer is much larger owing to the sparsity of data. Based on very uncertain information, the reinsurer has to decide:

  - whether it wants to write the risk at all;
  - what price it should charge for a given risk, if it is the leader, or whether it wants to accept the leader's rate;
  - how big a line (percentage) it wants to write, given the constraints imposed by the insurance company.

- The structure of this simplified reinsurance market can be represented as a weighted bipartite graph, that is a 4-tuple $G = (U, V, E, W)$ where $U = (u_1, \dots u_m)$ and $V = (v_1, \dots v_n)$ are two sets of vertices representing the reinsurers and the insurers respectively, and E is a subset of $U \times V$ – that is, a set of edges which connect reinsurers and insurers. Unlike the personal insurance setting, the connection is many-to-many. Each edge $e_{i,j} = (u_i, v_j) \in U \times V$ is assigned a weight $w(e)$ which represents the line written by reinsurer j in a contract concerning insurer i. The weights are subject to the constraint that $\sum_j w(e_{i,j}) = 1$. A representation of this graph is shown in Figure 6.4.

- The other considerations made for the personal lines insurance market (the utility function, the role of the regulator, the different ability to price, etc) hold for this market as well.

With all these ingredients, one can try different strategies using stochastic simulation and tournaments, as for the personal insurance market. The objective is again to determine the optimal pricing strategy. The possibility of a reinsurer's default is here more important because insurance companies are not as protected as customers by the regulator in case of insolvency of the reinsurer.

To make this model more realistic, one can introduce:

- reinsurance brokers, which have their own preferred markets (reinsurers) and charge a commission;
- vertical pricing (where suitable, as in the aviation market);
- product differentiation (eg excess-of-loss reinsurance with aggregate deductible/limit, and other bells and whistles);
- stickiness to preserve relationship.

6.5 Practical insurance example: Co-insurance of environmental risks in Italy

The examples we have seen so far were strictly non-cooperative: each agent (customer, insurer or reinsurer) looked at its own utility as the exclusive guide to actions, and pricing decisions were taken in isolation, without consultation among its peers. In the case of insurers, consultation is of course theoretically possible but obviously illegal, as it amounts to forming a cartel.

For some problems, cooperative solutions are possible and legal, and analytical solutions based on game theory have been devised. An example can be found in the paper by Fragnelli and Marina, "A fair procedure in insurance" [FM03]. The problem investigated is that of the optimal co-insurance of a risk R with an overall premium ϕ among n companies. This approach follows a line research that goes back to a paper by Charnes and Granot [CG73].

A real-world example where this type of scenario applies is the co-insurance of environmental risks in Italy, where a pool of 61 insurance companies is responsible for all such risks and the overall price is set by the government [AFM06].

In order to find an acceptable solution, the authors require the allocation to be Pareto efficient, individually rational and fair. "Fair" is in this context defined as an envy-free allocation, i.e. no party would exchange its portion of the risk with that of someone else.

The optimal decomposition of a risk will reflect the different ways in which different companies estimate risk. The idea behind the solution given in [FM03] is that each company has to receive a quota of the premium equivalent to its evaluation of its quota of risk, plus a compensation that eliminates possible envy towards the other companies, and finally it receives a suitable amount of the residual of the premium. Notice that different companies will evaluate risk in a different way! If the companies wish to avoid exchanging information on how they evaluate risk, a mediator is required. As the envy-free solution leads to some excesses, alternative solutions are considered in [AFM06].

Also notice that despite the existence of “lines” as in reinsurance, this scenario does not apply to reinsurance as it operates in the London market, where the price is typically agreed between the broker and the lead underwriter and the others join in at that price (horizontal placement) or with their own price (vertical placement), but in a strictly sequential order, with no possibility of negotiation among the selling parties. In the co-insurance scenario we have discussed here, negotiation is possible and legal because the overall price is fixed a priori by the client, and therefore the client is not damaged by these negotiations.

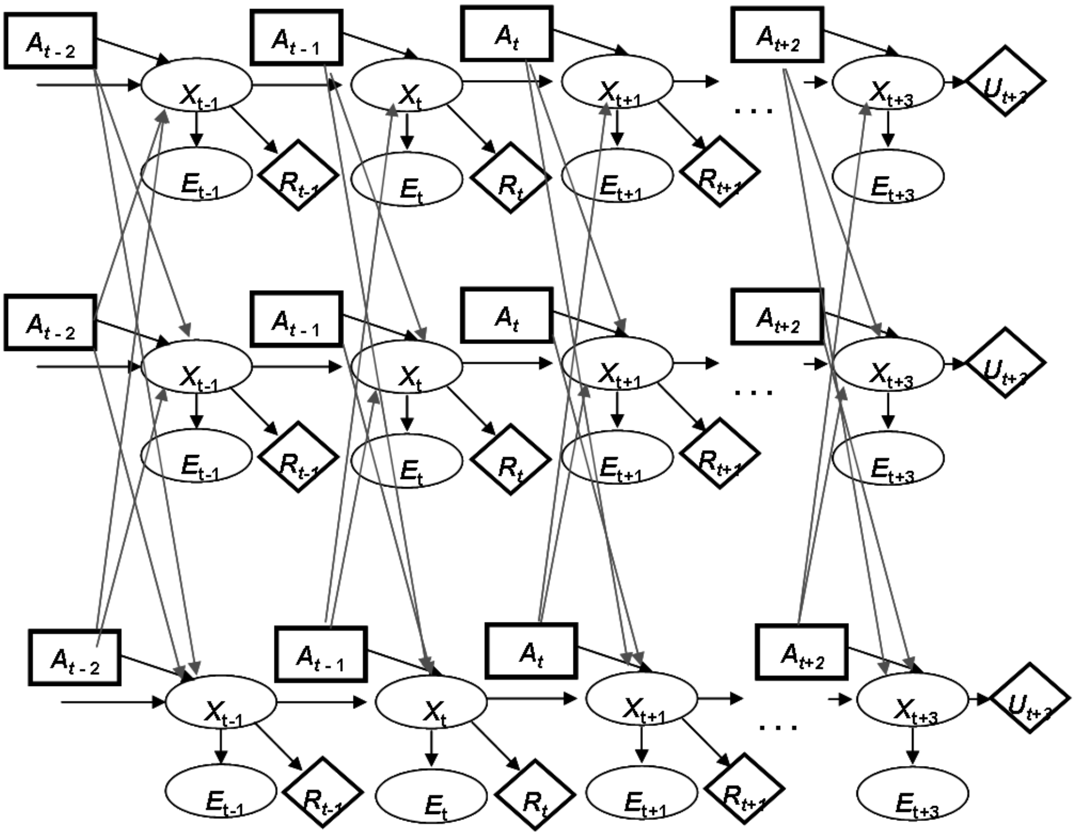


Figure 6.3: The personal lines market seen as a network of dynamic decision networks. Notice how the actions of one DDN (one insurance company) affect the actions of the other companies.

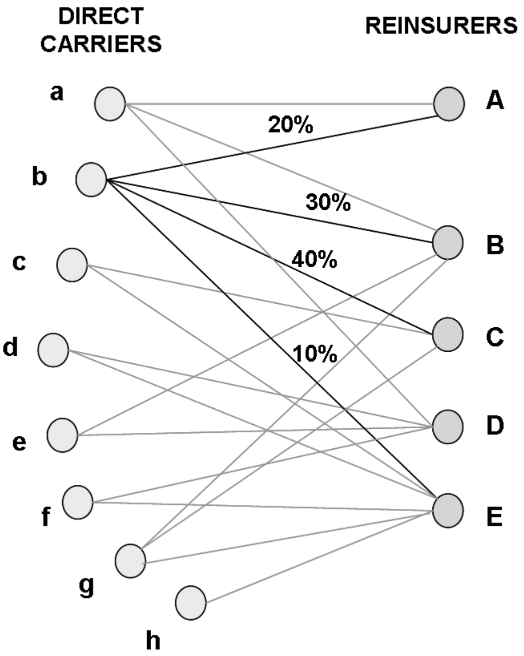


Figure 6.4: A sketchy representation of the reinsurance market, ignoring the role of brokers.

Figure 6.5: Muddy, messy and full of things that we don't understand.

Chapter 7

Conclusions

All beginnings are hard.
Chaim Potok, *In the beginning*

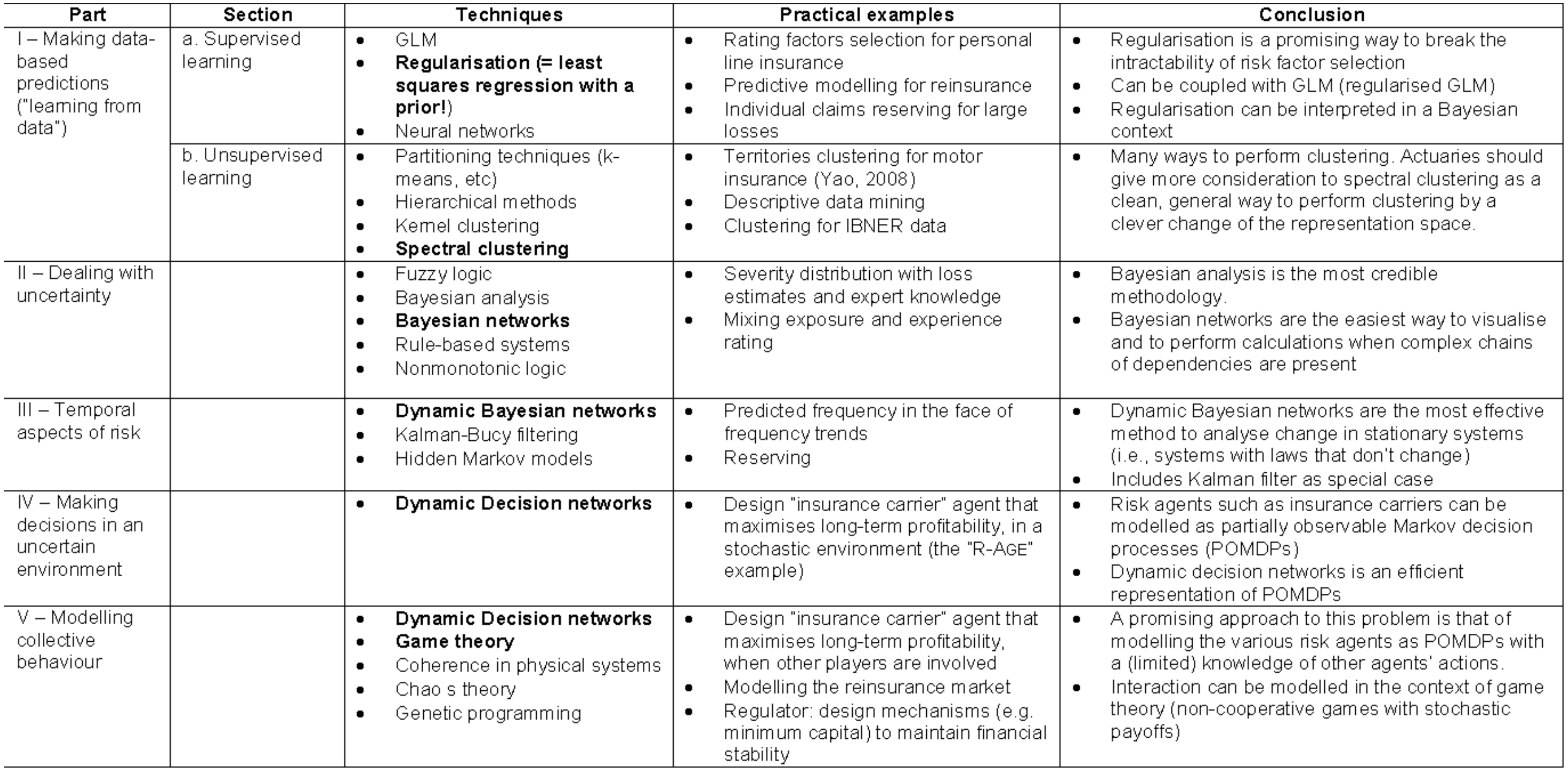
It is now time to draw the conclusions of this investigation. Section 7.1 argues that the "risk agents" paradigm is currently the most promising framework for understading and describing risk. Some practical finding are presented in Section 7.2. The limitations of the "risk agents" paradigm and of this research itself are outlined in Section 7.3. Finally, the need for future research and an explanation where this could take us are highlighted in Section 7.4.

A one-page overview of the conclusions is available in Figure 7.1.

7.1 The "risk agent" paradigm

The conclusion of this investigation is that computational intelligence provides a natural framework for addressing a large range of actuarial problems. The author's main concern – that this might mean trading the rigour of classical statistics with a discipline with shaky foundations – was assuaged after observing the degree to which data-driven machine learning has become rigorous, thanks to the effort of many leading statisticians such as those of the Stanford school, and how much the treatment of agents in stochastic environment (especially in terms of partially observable Markov decision processes) has progressed.

More specifically, these are the highlights of this investigation:



Part	Section	Techniques	Practical examples	Conclusion
I – Making data-based predictions ("learning from data")	a. Supervised learning	• GLM • **Regularisation (= least squares regression with a prior!)** • Neural networks	• Rating factors selection for personal line insurance • Predictive modelling for reinsurance • Individual claims reserving for large losses	• Regularisation is a promising way to break the intractability of risk factor selection • Can be coupled with GLM (regularised GLM) • Regularisation can be interpreted in a Bayesian context
	b. Unsupervised learning	• Partitioning techniques (k-means, etc) • Hierarchical methods • Kernel clustering • **Spectral clustering**	• Territories clustering for motor insurance (Yao, 2008) • Descriptive data mining • Clustering for IBNER data	• Many ways to perform clustering. Actuaries should give more consideration to spectral clustering as a clean, general way to perform clustering by a clever change of the representation space.
II – Dealing with uncertainty		• Fuzzy logic • Bayesian analysis • **Bayesian networks** • Rule-based systems • Nonmonotonic logic	• Severity distribution with loss estimates and expert knowledge • Mixing exposure and experience rating	• Bayesian analysis is the most credible methodology. • Bayesian networks are the easiest way to visualise and to perform calculations when complex chains of dependencies are present
III – Temporal aspects of risk		• **Dynamic Bayesian networks** • Kalman-Bucy filtering • Hidden Markov models	• Predicted frequency in the face of frequency trends • Reserving	• Dynamic Bayesian networks are the most effective method to analyse change in stationary systems (i.e., systems with laws that don't change) • Includes Kalman filter as special case
IV – Making decisions in an uncertain environment		• **Dynamic Decision networks**	• Design "insurance carrier" agent that maximises long-term profitability, in a stochastic environment (the "R-AGE" example)	• Risk agents such as insurance carriers can be modelled as partially observable Markov decision processes (POMDPs) • Dynamic decision networks is an efficient representation of POMDPs
V – Modelling collective behaviour		• **Dynamic Decision networks** • **Game theory** • Coherence in physical systems • Chao s theory • Genetic programming	• Design "insurance carrier" agent that maximises long-term profitability, when other players are involved • Modelling the reinsurance market • Regulator: design mechanisms (e.g. minimum capital) to maintain financial stability	• A promising approach to this problem is that of modelling the various risk agents as POMDPs with a (limited) knowledge of other agents' actions. • Interaction can be modelled in the context of game theory (non-cooperative games with stochastic payoffs)

Figure 7.1: An overview of the conclusions.

- The familiar problems of pricing, reserving and capital modelling based on data and soft knowledge are most naturally viewed as examples of *supervised machine learning.* Common techniques include (regularised) regression, GLM, neural networks.

- Exploratory analysis can benefit from unsupervised learning techniques such as clustering and data mining.

  - This is crucial especially for personal lines insurance, where data mining offers the possibility of finding hidden patterns in large data sets.
  - Also, clustering can be used to identify a small number of values for risk factors that are multi-dimensional in nature.

- Uncertain and soft knowledge can be dealt with most successfully in a Bayesian context.

  - Although other approaches (fuzzy set theory, non-monotonic logic) are possible, the Bayesian approach seems to the author to be the most suited to the highly numerical insurance context, whereas fuzzy set theory and non-monotonic logic are better suited to deal with linguistic vagueness.
  - When decisions are made based on complex, multi-stage processes, Bayesian networks can help propagate beliefs and uncertainties.

- The fact that risk changes over time and that the information on past risk also unfolds gradually can be addressed by probabilistic reasoning over time, which allows the evolution of the parameters and the structure of a model. This formalism is often referred to as the state-space model. Problems that can be naturally framed in this context include reserving and pricing in the face of a changing risk profile.

- Agents – such as insurers – making decisions in an uncertain environment can be seen as Markov decision processes. The aim of risk agents is to devise optimal strategies. A discounted utility can be associated with each state and with each strategy based on the future rewards. An optimal strategy is that which maximises the *expected* utility. Of particular interest to risk are the so-called partially observable Markov decision processes (POMDPs), as they assume that the state in which the agent is is not known for certain. This reflects the notion that insurance companies do not usually know their current financial position with certainty, especially if their portfolio has long-tail elements.

- The environment cannot always be considered a "blur" – it contains other agents whose behaviour cannot be ignored if we are to make successful decisions. Risk agents, in fact, interact with each other by competing, cooperating, exchanging goods and information, forming what is called a *multi-agent system.* Game theory then provides a framework for modelling the interaction between agents. In most cases, an analytical treatment of the problem in terms of game theory will not be feasible, and stochastic simulation will be the main tool by which one tries strategies and responds to other insurers' conjectured strategies. Stochastic game theory settings such as that by Axelrod [Axe84] will help addressing both the problem of *agent design* (by which the optimal policy for an agent is selected) and the problem of *mechanism design*, by which a regulator finds rules that maximise the overall utility of the market, however it is defined.

Does this actually answer the question "What is the appropriate framework for describing and understanding risk?". It certainly does not provide a full answer, in the sense that we have not proved in any rigorous way that this is the appropriate framework. Such a thing would be impossible to prove anyway. What we have shown is that most actuarial problems can be framed quite naturally in the computational intelligence context, and that by doing so we have a number of results from computational intelligence that we can tap into. This investigation also shows that this framework is more adequate than the current framework, which can be basically summarised as 'classical statistics enhanced by professional judgment'. It is superior in the sense that it provides a practical way to incorporate uncertainty reasoning and judgment into decision-making. This does not mean at all that human judgment can be replaced – rather it is a way of incorporating it in a transparent way, forcing us to declare where judgment is used, how uncertain that itself is, and providing a methodology to revise it in the face of better evidence.

7.2 Practical findings

Although the main question addressed by this paper was epistemological, "What is the appropriate framework to describe and understand risk?", this paper contains a number of practical results:

- A comparison of different statistical learning techniques was carried out. The techniques looked at were GLM (the current industry standard), neural networks and regularisation. We have found that neural networks – in themselves a rigorous non-linear statistical tool – are of limited use for general insurance applications as they provide prediction without interpretation: in other terms, they calculate risk without enhancing our understanding of it. One possible use of neural networks, however, is as a benchmark – if the expected prediction error obtained with a neural network is significantly smaller than that obtained with other techniques, this may be a sign that we have to change the structure of our model or the dictionary of functions we are using.

Regularised regression provides a valid alternative to GLM, and sparsity-based techniques such as the lasso and the elastic net are especially interesting as they provide an efficient way of selecting the relevant variables. Elastic net should also be considered for use with situations in which the number of observations is limited, even far smaller than the number of parameters.

The main drawback of regularised regression is that usually this uses a squared loss function, which leads to a deterioration of results in the case, for example, of Poisson noise. GLM uses a $\log P$ loss function which is more general. The most promising approach is probably regularised GLM, where one can use a more general loss function ($\log P$) and still benefit from the application of sparsity-based techniques.

- Clustering techniques (also from machine learning) are useful for a number of tasks ranging from exploratory analysis to data mining. When addressing these problems, the actuarial community should be aware not only of the basic clustering techniques such as K-means but also of more modern approaches such as spectral clustering, which provide far more flexibility.

- Machine learning also offers a number of different methods for model selection and diagnostics for model validation: splitting the observations into three sets (training sample, selection sample, test sample), k-fold cross-validation, AIC, BIC, MDL... Some of these diagnostics (hold-out sample, AIC) are familiar to the actuarial community, especially in personal lines, but we would benefit from enlarging our toolkit, especially with the systematic use of k-fold cross-validation, which provides a clever way of overcoming the problem of scarce data.

- Data uncertainty can be solved naturally in a Bayesian context, provided we have a probabilistic model of how the actual estimates relate to the true estimates. As there are usually multiple sources of data uncertainty at different stages of the modelling process, Bayesian networks provide a useful way to calculate efficiently the relevant probabilities.

- The problem of integrating different sources of knowledge also finds a natural solution in the Bayesian context, and again Bayesian networks are especially useful. Our simple example of the calculation of the parameters of a severity distribution with data uncertainty and prior knowledge of the parameters is an example of how this can be performed with a Bayesian network.

- Bayesian networks can be easily generalised to deal with information that changes with time (dynamic Bayesian networks) and to incorporate the decision process (dynamic decision networks). Dynamic decision networks can be designed to compete with each other in a multi-agent system, allowing to analyse the collective behaviour of markets.

7.3 Limitations

The main limitation of computational intelligence techniques is that they themselves capture the ecological element only up to a point:

- There is a hidden assumption in many applications of machine learning – that the environment is stable. For example, we make sure that the inferences we draw from our data are sound by putting aside a subset of these data and testing our theory (based on a training set) against this subset (see Section 2.1.2). However, this may not tell us much of what will happen when we use our theory on data that have not been collected yet – e.g., losses that have yet to happen. This is a crucial difficulty for us in that risk is ever-changing. Since we have to do these predictions anyway, these will have to be based on our "sniffing out" the environment and using this soft knowledge the best we can, in the usual Bayesian framework.

- One of the reasons why risk is ever-changing is that changes in the environment are often introduced by people. Especially in finance, people work the system constantly and introduce changes that are difficult to model almost by definition. This again is a perfectly legitimate ecological aspect of risk, and one for which current techniques are inadequate. To solve this problem, we would need what is usually called "strong artificial intelligence" (that is, agents that actually exhibit general intelligence, much like humans). This is, despite some claims to the contrary, a highly speculative branch of computational intelligence, with little to show for in terms of practical achievements.

- Despite the fact that multi-agent systems are a very promising paradigm for collective behaviour, a lot more research is needed before they can actually be used for taking optimal decisions at an individual and at the regulator's level. The sheer model complexity may prevent them *ever* to be used as optimal decision tools: however, they would still be useful in the production of unexpected scenarios.

Alongside the limitations of the computational intelligence discipline, there are many limitations of this research itself, which should be pointed out in the spirit of not blaming one's tools for the possibly failed carpentry:

- the treatment of the different topics is obviously uneven: the section on learning from data is perhaps hypertrophic, whereas much more research would be needed to investigate probabilistic reasoning in time and multi-agent systems. This reflects both how much machine learning is developed with respect to, say, the theory of partially observable Markov decision processes. However, it also reflects the fact that most researchers with whom I had the opportunity to exchange ideas and who could introduce me to cutting-edge, sometimes unpublished, ideas belonged to the machine learning community;
- depth of coverage was sacrificed to breadth. Each section, if developed properly, would be a large project on its own. All the practical examples are by necessity very simple, and in some cases it was not possible fully to develop worked-out numerical examples in a reasonable time frame (for example, when analysing collective behaviour);
- even though depth was sacrificed to breadth, the breadth of coverage itself could be much improved and plenty of material had to be left out. Kernel techniques for supervised and unsupervised learning have been excluded; genetic algorithms have been cited only in passing; projection pursuit (a supervised learning technique, akin to neural networks) has not been mentioned; boosting techniques have not been mentioned, either; little emphasis has been given to numerical techniques for Bayesian analysis, such as MCMC and Gibbs sampling (although the reason is, in this case, that these are very well-known and

adequate treatment can be found in any book on Bayesian analysis); reinforcement learning has not been addressed; and so on;

- naturally, there are many techniques in computational intelligence that do not find a natural application to general insurance. This might have left the impression in the reader that this paper was trying to prove that there is a one-to-one correspondence between the two disciplines. This was not the aim of this paper, and it should be stressed that there is much else in computational intelligence that has been left out just because it did not seem useful enough to me: for example, constraint satisfaction is an important technique of computational intelligence and although the concept finds a place in general insurance (for example, finding a reinsurance structure which satisfies many different constraints), the size of the problem is simply not large enough to need a formal approach. On the other hand, there are of course many important topics in general insurance that have nothing to do with computational intelligence (most legal aspects, for example). Finally, there is certainly much which has been left out simply for lack of imagination on the author's side!

7.4 Future research

The limitations of this investigation listed above naturally suggest some spin-offs in terms of future research. To list but a few:

- although the main points of the comparison between GLM and regularised regression have been touched upon, this comparison should be made far more systematic. This could be achieved by a battery of controlled experiments for the most common problems encountered in predictive modelling (frequency modelling, severity modelling, aggregate loss modelling). Regularised GLM should also be tested to see how it performs with respect to regularised (squared loss) regression and traditional GLM;

- a more systematic comparison of different clustering techniques on IBNER data should also be attempted. The example analysed here only involved five years of development for data, but the most general case should be analysed, possibly on a larger set of data. It should also be determined whether this helps improving the results of predictive modelling for IBNER purposes or not;

- some actual stochastic simulations should be attempted to model collective behaviour. It would actually be interesting to involve a working party in this sort of effort, with different participants providing different strategies and having a tournament among these strategies, in the spirit of Axelrod's experiment. Both the "mechanism design" and the "agent design" problem should be tackled.

Bibliography

[AFM06] Daniela Ambrosino, Vito Fragnelli, and Maria Marina. Resolving an insurance allocation problem: A procedural approach. *Social Choice and Welfare*, 26(3):625–643, June 2006.

[Axe84] Robert Axelrod. The evolution of cooperation. 1984.

[Ben96] Edward A. Bender. *Mathematical Methods in Artificial Intelligence*. IEEE Computer Society, 1996.

[BHMM09] John Berry, Gary Hemming, Georgy Matov, and Owen Morris. Report of the model validation and monitoring in personal lines pricing working party. *GIRO*, 2009.

[Bis07] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer, 2007.

[Buc83] James J. Buckley. Decision making under risk a comparison of bayesian and fuzzy set methods. *Risk Analysis*, 3(3), 1983.

[CD97] J. David Cummins and Richard A. Derrig. Fuzzy financial pricing of property-liability insurance. *North American Actuarial Journal*, 1(4):21–44, 1997.

[CG73] A. Charnes and D. Granot. Prior solutions: extensions of convex nucleolus solutions to chance-constrained games. *Proceedings of the COmputers Science and Stochastics Seventh Symposium at Iowa State University*, pages 323–332, 1973.

[DBC$^+$03] C. Dugas, Y. Bengio, N. Chapados, P. Vincent, G. Denoncourt, and C. Fournier. Statistical learning algorithms applied to automobile insurance ratemaking, 2003.

[DMDVR09] Christine De Mol, Ernesto De Vito, and Lorenzo Rosasco. Elastic-net regularization in learning theory. *J. Complex.*, 25(2):201–230, 2009.

[EHJT04] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.

[FM03] Vito Fragnelli and Maria Erminia Marina. A fair procedure in insurance. *Insurance: Mathematics and Economics*, 33(1):75–85, 2003.

[FM05] Vito Fragnelli and Maria Erminia Marina. Subscribing an insurance policy is gambling? *Atti del X Convegno di Teoria del Rischio - Campobasso*, pages 153–160, 2005.

[GMP05] Peter D. Grünwald, Jae Myung, and Mark A. Pitt. *Advances in minimum description length.* MIT Press, 2005.

[Guo03] Lija Guo. Applying data mining techniques in property/casualty insurance. *CAS Forum*, 2003.

[Has] Trevor Hastie. Regularization paths and coordinate descent.

[HK09] Gareth Haslip and Colin Kerley. Optimisation in a capital scarce world. *Presentation at the GIRO Conference, Edinburgh*, 2009.

[HKP91] John. Hertz, Anders. Krogh, and Richard G. Palmer. *Introduction to the theory of neural computation.* Addison-Wesley Pub. Co., Redwood City, Calif. :, 1991.

[HTF01] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning.* Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[InT] In the night garden. *BBC Series.*

[JZ83] Piet De Jong and Ben Zehnwirth. Claims reserving, state-space models and the kalman filter. *JIA*, (110):157–181, 1983.

[Lem90] J. Lemaire. Fuzzy insurance. *Astin Bulletin*, 20(1):33–55, 1990.

[LL09] Qing Li and Nan Lin. The bayesian elastic net. *To be published*, 2009.

[LP96] Julian Lowe and Louise Pryor. Neural networks v. glms in pricing general insurance. *General Insurance Convention*, 1996.

[MFS$^+$04] Claudine Modlin, Sholom Feldblum, Doris Schirmacher, Ernesto Schirmacher, Neeza Thandi, and Duncan Anderson. A practitioner's guide to generalized linear models. *CAS Discussion Paper*, 2004.

[MMTV] C. De Mol, S. Mosci, M. Traskine, and A. Verri. A regularized method for selecting nested group of genes from microarray data. *Journal of Computational Biology (to appear).*

[MN90] P. McCullagh and J. A. Nelder. *Generalized Linear Models.* CRC Press, 2nd edition, 1990.

[MP43] W. S. Mcculloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5:115–133, 1943.

[MP69] M. L. Minsky and S. A. Papert. *Perceptrons.* Cambridge: MIT Press, 1969.

[NCC99] Efstratios Nikolaidis, Harley H. Cudney, and Q. Chen. Bayesian and fuzzy set methods for design under uncertainty, 1999.

[OPT98] M. R. Osborne, B. Presnell, and B. A. Turlach. Knot selection for regression splines via the lasso. *Computing Science and Statistics*, 30:44–49, 1998.

[Pap94] Christos M. Papadimitriou. *Computational complexity.* Addison-Wesley, Reading, Massachusetts, 1994.

[Pea88] Judea. Pearl. *Probabilistic reasoning in intelligent systems : networks of plausible inference.* Morgan Kaufmann Publishers, San Mateo, Calif., 1988.

[PH96] Mee Young Park and Trevor Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society. Series B (Methodological)*, 69(4):659–677, 1996.

[Pop59] Karl Popper. The propensity interpretation of probability. *The British Journal for the Philosophy of Science*, X(37):25–42, 1959.

[RDVC$^+$04] Lorenzo Rosasco, Ernesto De Vito, Andrea Caponnetto, Michele Piana, and Alessandro Verri. Are loss functions all the same? *Neural Comput.*, 16(5):1063–1076, 2004.

[RHM87] D. E. Rumelhart, G. E. Hinton, and J. L. McClelland. A general framework for parallel distributed processing. In D. E. Rumelhart, J. L. McClelland, et al., editors, *Parallel Distributed Processing: Volume 1: Foundations*, pages 45–76. MIT Press, Cambridge, 1987.

[RN03] S. J. Russell and Norvig. *Artificial Intelligence: A Modern Approach (Second Edition).* Prentice Hall, 2003.

[Ros58] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6):386–408, November 1958.

[Sha04] Arnold F. Shapiro. Fuzzy logic in insurance. *Insurance: Mathematics and Economics*, 35(2):399–424, 2004.

[Tay08] Greg Taylor. A simple model of insurance market dynamics. *Proceedings of the GIRO Conference, Sorrento*, 2008.

[Tib96] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[TTG94] Demetri Terzopoulos, Xiaoyuan Tu, and Radek Grzeszczuk. Artificial fishes: autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artif. Life*, 1(4):327–351, 1994.

[vL07] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[Woo01] Michael J. Wooldridge. *Multi-agent systems : an introduction.* Wiley, Chichester, 2001. GBA1-Z6596 Michael Woolridge.

[Yao08] Ji Yao. Clustering in ratemaking: with application in territories clustering. *Casualty Actuarial Society Discussion Paper Program Casualty Actuarial Society - Arlington, Virginia*, pages 170–192, 2008.

[ZH05] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67:301–320, 2005.

[ZO04] Franco Zambonelli and Andrea Omicini. Challenges and research directions in agent-oriented software engineering. *Autonomous agents and multi-agent systems*, 9(3):252–283, 2004.