Università degli Studi di Trieste

Dipartimento di Scienze Economiche, Aziendali, Matematiche e Statistiche

Corso di Laurea Magistrale in Scienze Statistiche e Attuariali

Tesi di Laurea Magistrale

Application of GLM Advancements to Non-Life Insurance Pricing

Candidato:

Leonardo Stincone

Relatore:

Prof. Francesco Pauli

Anno Accademico 2019 - 2020

The data scientist is a person
who is better at statistics than any software engineer
and better at software engineering than any statistician.

Josh Wills

Table of Contents

List of Figures

List of Tables

List of Acronyms

CPU Central Processing Unit

ETL Extract Transform Load

GAM Generalized Additive Model

GBM Gradient Boosting Machine

GLM Generalized Linear Model

HaaS Hardware as a Service

HDD Hard Disk Drive

IBNeR Incurred But Not enough Reported claim

IBNyR Incurred But Not yet Reported claim

IT Information Technology

LASSO Least Absolute Shrinkage and Selection Operator

MTPL Motor Third Party Liability

NN Neural Network

RAM Random Access Memory

RF Random Forest

SSD Solid State Drive

Introduction

La mia introduzione ...

Thesis aim

Lorem ipsum ...

Actuary and datascientist figure

Lorem ipsum ...

Thesis structure

Lorem ipsum ...

CHAPTER 1

Non-Life Insurance Pricing

In this chapter we are going to provide an overview on how non-life insurance works from an actuarial point of view with a specific focus on the retail pricing process.

1.1 What a Non-Life Insurance is

The Italian Civil Code [1] provides the following definition of insurance contract:

Definition 1.1 (Insurance Contract, Art. 1882, Italian Civil Code)**.** The insurance is the contract by which an insurer, in exchange of the payment of a certain premium, obliged himself, within the agreed limits:

1. to pay an indemnity to the insured equivalent to the damage caused by an accident;
2. or to pay an income or a capital if a life-related event occurs.

This definition identifies two parties: the *Insurer* and the *Policyholder*. The policyholder pays to the Insurer a certain *Premium* at the beginning of the insurance coverage and the insurer will pay a benefit if a certain event (*Claim*) occurs. This event could happen zero, one or more than one times, so it is possible to have more than one claim.

Usually, in non-life insurance, the benefit is the payment of a sum. This sum could be predetermined (e.g. in motor theft insurance, where the benefit is usually the value of the insured vehicle) or defined by the entity of the claim (e.g. in Motor Third Party Liability (MTPL) insurance, it depends on the damage the policyholder has caused to a third party). Regarding the "agreed limits", another peculiarity of non-life insurances is that the coverage period is defined as a fixed amount of time, usually corresponding to 1 year.

Starting from this legal definition, we can formalize a non-life insurance contract as follows.

Let:

- $]t_1, t_2]$, with $t_1 < t_2$, be the coverage period;
- $P > 0$ be the premium paid by the policyholder to the insurer;
- $N \in \mathbb{N}$ be the number of claims occurred during the coverage period (*claims count*);
- $\tau_1, \tau_2, \dots, \tau_N$, with $t_1 < \tau_1 < \tau_2 < \cdots < \tau_N < t_2$, be the timing of each claim;
- $Z_1, Z_2, \dots, Z_N > 0$ be the amount of each claim (*claims severities* or *claims sizes*).

The total cost of claims for the insurance is

$$S = \begin{cases} 0 & \text{if } N = 0 \\ \sum_{i=1}^{N} Z_i & \text{if } N > 0 \end{cases}$$

.

For simplicity, in the following we are going to just use the notation $S = \sum_{i=1}^{N} Z_i$ with the meaning of 0 if $N = 0$.

Figure 1.1 shows the cash flows corresponding to the insurance contract. From this representation we can interpret the entering into an insurance contract by the policyholder as a way to exchange the negative cash flows $-Z_1, -Z_2, \dots, -Z_N$ with one single negative cash flow $-P$. On the other hand, the insurer undertakes the negative cash flows $-Z_1, -Z_2, \dots, -Z_N$ in exchange for a positive cash flow $+P$.

The major difference between these cash flows is that P is a certain amount, while $Z_1, Z_2, \dots, Z_N$, at the time t_1, are uncertain in the amount, in the count (N) and in the timing ($\tau_1, \tau_2, \dots, \tau_N$). So, the policyholder, paying a premium P, is giving his risk to the insurer.

This representation points out the inversion of the production cycle typical of the insurance activity. From the insurer point of view, the revenue emerges at the beginning of the economic activity, in t_1, while the costs will emerge later. In most other economic activities, the costs emerge before the selling of the product, so the agent can choose the selling price taking into account how much that product costed him. In insurance activity, the insurer, when is selling his product (the insurance coverage), doesn't know the amount of claims he is going to pay for that product. Thus, it is crucial to properly predict the future costs in order to determine an adequate premium.

From a statistical point of view, we can translate this uncertainty saying that N and $Z_1, Z_2, \dots, Z_N$ are random variables. Therefore, we can say that $\{N, Z_1, Z_2, \dots\}$ is a stochastic process. Usually, in non-file insurance pricing, the variables $\tau_1, \tau_2, \dots, \tau_N$ are not taken into account because the coverage span is short and from a financial point of view the timing of the claims occurrences has negligible effect.

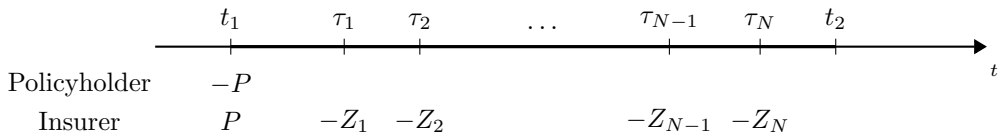


Figure 1.1: Insurance Contract cash flows.

Previously we said that $Z_1, Z_2, \ldots, Z_N$ are all positive. This assumption corresponds to the fact that we are excluding the null claims, i.e. the claims that have been opened, but result in no payment due by the insurer. For the values of Z_i with $N < i$ we can use the rule that $\{N < i\} \Rightarrow \{Z_i = 0\}$, so $Z_{N+1} = 0$, $Z_{N+2} = 0$, ... Therefore, we can say that:

$$\{N < i\} \Longleftrightarrow \{Z_i = 0\}$$

1.2 Non-Life insurance pricing

In insurances, the premium that the the insurer offers to the policyholder in exchange for the insurance coverage is not the same for every policyholder. The insurer evaluates the risk related to that policy and determine a "proper" premium taking into account risk related factors and commercial related factors. The process of *pricing* corresponds in defining the set of rules for determining this "proper" premium P_i for a specific policyholder i, given the known information on him. In the next sections we are going to better explain what "proper" means.

1.2.1 Compound distribution hypotheses

The first step for evaluating the stochastic process $\{N, Z_1, Z_2, \ldots\}$ is to introduce some probabilistic hypotheses. The usual hypotheses assumed are the following:

Definition 1.2 (Compound distribution)**.** Let's assume that:

1. for each $n > 0$, the variables $Z_1|N = n$, $Z_2|N = n$, ..., $Z_n|N = n$ are stochastically independent and identically distributed;
2. the probability distribution of $Z_i|N = n$, $i \leq n$ does not depend on n.

Under these hypotheses we say that:

$$S = \sum_{i=1}^{N} Z_i$$

has a compound distribution.

The variable $Z_i|N = n$ used in this definition can be interpreted as the *claim severity for the i^{th} claim under the hypothesis that n claims occurred*. The two hypotheses provided in definition 1.2 imply that the distribution of $Z_i|N = n,\ i \leq n$ does not depend on i nor on n. For this reason, in the following, we are going to use the notation Z to represent a random variable with the ditribution of $Z_i|N = n,\ i \leq n$ and $F_Z(\cdot)$ for its cumulative distribution function (i.e. $F_Z(z) = P(Z \leq z)$).

Let's consider the variabile $Z_i|N \geq i$. We can interpret it as the *claim severity for the i^{th} claim under the hypothesis that the i^{th} claim occurred*. From the hypotheses provided in definition 1.2 we can obtain that also $Z_i|N \geq i$ has the same distribution of $Z_i|N = n,\ i \leq n$. This can be easily obtained as follows:

$$
\begin{aligned}
P\left(Z_i \leq z | N \geq i\right) &= P\left(Z_i \leq z \middle| \bigvee_{n=i}^{+\infty} (N = n)\right) && (1.1)\\
&= \sum_{n=i}^{+\infty} \underbrace{P\left(Z_i \leq z | N = n\right)}_{=F_Z(z)} P\left(N = n | N \geq i\right) && (1.2)\\
&= \sum_{n=i}^{+\infty} F_Z(z) P\left(N = n | N \geq i\right) && (1.3)\\
&= F_Z(z) \underbrace{\sum_{n=i}^{+\infty} P\left(N = n | N \geq i\right)}_{=1} && (1.4)\\
&= F_Z(z)
\end{aligned}
$$

Where:

- the step (1.1) and the step (1.2) are given by the fact that the event $\{N \geq i\}$ can be decomposed as $\{N \geq i\} = \left\{\bigvee_{n=i}^{+\infty} (N = n)\right\}$ and that the events $\{N = n\}, n \in \{i, i+1, i+2, \dots\}$ are two-by-two disjoint, so they constitute a partition of $\{N \geq i\}$, that allows us to use the disintegrability property of the probability;
- the step (1.3) is due to the fact that the distribution of $Z_i \leq z | N = n$ depends neither on i nor on n;
- the equivalence $\sum_{n=i}^{+\infty} P\left(N = n | N \geq i\right) = 1$ at step (1.4) is due to the fact that the events $\{N = n\}, n \in \{i, i+1, i+2, \dots\}$ are a partition of $\{N \geq i\}$.

Therefore, Z can be considered as the *claim severity for a claim under the hypothesis that that claim occurred*.

1.2.2 Distribution of the total cost of claims

Under the hypotheses of definition 1.2, it is possible to obtain the full distribution of S given the distribution of N and Z. In this chapter we are going to provide

only the formula of the expected value $E(S)$, but, with the same approach one can obtain all the moments.

The expected value of the total cost of claims $E(S)$ can be obtained from the expected value of the claims count $E(N)$ and the expected value of the claim severity $E(Z)$ as follows:

$$
\begin{aligned}
E(S) &= \sum_{n=0}^{+\infty} P(N=n)\, E\left(S|N=n\right) && (1.5)\\
&= \sum_{n=0}^{+\infty} P(N=n)\, E\left(\sum_{i=1}^{n} Z_i \middle| N=n\right) && (1.6)\\
&= \sum_{n=0}^{+\infty} P(N=n) \sum_{i=1}^{n} \underbrace{E\left(Z_i|N=n\right)}_{=E(Z)} && (1.7)\\
&= \sum_{n=0}^{+\infty} P(N=n) n E(Z) && (1.8)\\
&= E(Z) \underbrace{\sum_{n=0}^{+\infty} n P(N=n)}_{=E(N)} && (1.9)\\
&= E(N)E(Z) && (1.10)
\end{aligned}
$$

Where:

- the step (1.5) is given by the fact that the events $\{N=0\}, \{N=1\}, \{N=2\}, \ldots$ constitute a partition of the certain event Ω, that allows us to use the disintegrability property of the expected value;
- the step (1.6) is due to the definition of S;
- the step (1.7) is due to the linearity of the expected value;
- the steps (1.8) and (1.9) are due to the fact that, as assumed by the compound distribution hypotheses, $E\left(Z_i|N=n\right)$ does not depends on i and n;
- the step (1.10) is due to the definition of the expected value $E(N) = \sum_{n=0}^{+\infty} nP(N=n)$.

This result tells us that, under the hypotheses of the compound distribution, it is possible to easily obtain $E(S)$ from $E(N)$ and $E(Z)$. That means that we can model separately $E(N)$ and $E(Z)$ and, from them, obtain $E(S)$. That result is particularly useful in personalization (paragraph 1.3), because, for each individual i, given the information we have on him, we can estimate his expected claim size $E(N_i)$ and his expected claim severity $E(Z_i)$ and obtain his expected total cost of claims as $E(S_i) = E(N_i)E(Z_i)$.

1.2.3 Risk premium and Technical Price

The expected cost of claims $E(S)$ is important because it gives us a first interpretation of what "proper" premium means.

Definition 1.3 (Risk Premium)**.** Said S the total cost of claims of a policyholder, his *Risk Premium* is given by:

$$P^{(risk)} = E(S)$$

The *Risk Premium* is the premium that on average covers the total cost of claims. As mentioned above, as the coverage spans are usually short, we are not taking into account the timing of the claims so we don't discount the fact that the claims occur later than the premium payment.

It is clear that this premium, that only covers the cost of claims, is not "proper" in the practice.

First of all, the insurer has to cover also the expenses related to the policy (commission on sales and expenses related to the claim settlement) and the general expenses of the company. Adding the expenses, we obtain the *Technical Price*.

Definition 1.4 (Technical Price)**.** Said S the total cost of claims of a policyholder and E the expenses related to his policy, his *Technical Price* is given by:

$$P^{(tech)} = E(S) + E = P^{(risk)} + E$$

Secondly, even if the policyholder paid a premium that on average covers claims and expenses, undertaking that risk with nothing in return would not make sense for the insurer. So, to the Technical Price, some further loadings must be added, as for example risk margin and profit margin.

The amount of the Technical Price with these loadings can be further modified based on business logic, as we are going to discuss later.

1.3 Modeling and Personalization

In this section we are going to better explain how pricing based on policyholder information works.

1.3.1 Pricing variables

Usually for every policyholder we have a certain amount of information on him that is considered relevant for his risk evaluation. This information must be reliable and observable at the moment of the underwriting of the policy.

In motor insurances, this information could be:

- Information on the insured vehicle: make, model, engine power, vehicle mass, age of the vehicle;
- General information of the policyholder: age, sex, address (region, city, postcode), ownership of a private box where he parks the car;
- Insurance specific information of the policyholder: number of claims caused in the previous years, how long he has been covered, bonus-malus class;
- Policy options: amount of the maximum coverage, presence and amount of a deductible, presence of other insurance guarantees, how many drivers will drive the vehicle;
- Customer information on the policyholder: how many years he has been a customer of the insurer, how many other policies he owns.
- Telematic data: how many kilometers per year the policyholder travelled in the previous years, which kind of roads the policyholder travelled on, the speed maintained during the trips, how many times the policyholder exceeded the speed limit, how many sharp accelerations and decelerations per kilometer the policyholder performed.

These pieces of information are usually called *pricing variables*.

We must observe that some of these variables are available for every potential customer (such as his age and address), while others are only available for policyholder that are already customers (such as telematic data that is available only if the policyholder agreed on installing on their car the device that collects this data).

Moreover, even considering the variables that are available for every customer, it is important to be aware of how reliable they are. Some of them come from official documents (as customer age and address or bonus-malus class), but others could be declared by the customer and his statements are not easily verifiable by the insurer (as the ownership of a private box or how many drivers will drive the vehicle).

The topic of variables reliability fits in the wider framework of fraud detection. Insurance companies put a lot of effort in preventing frauds. This is done with active actions, such as documents checks and inspections, and with predictive fraud detections models. The two most common categories of frauds are underwriting frauds (such as false declaration on insurance related data) and settlement frauds (such as faking an accident). The customer information on the policyholder is usually important to predict both underwriting frauds and settlement frauds. Usually customers that have a longer relationship with the company and own many policies are less likely to commit frauds.

Regarding the topic of variables reliability, the Italian Insurance Associations (ANIA) in the last years made some big steps forward by collecting in its databases a lot of information about policyholders and vehicles and making it available to insurance companies. For example, by logging in these databases it is possible, at the moment of the quote request, to retrieve useful insurance specific information such as the number of claims caused by the customer in the previous years or how long he has been covered and useful information on his vehicle such as when it has been registered or how many

changes of ownership did it experienced.

One of the roles of the actuary is to understand how reliable the information on the policyholder is and to decide how to use that information.

1.3.2 Pricing variables encoding

Formally the pricing variables can be encoded as a vector of real numbers. $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathcal{X} \subseteq \mathbb{R}^p$. In the modeling framework they can be also called explanatory variables, covariates, predictors or features.

The pricing variables can be of two types:

1. *Quantitative variables*: variables, like policyholder age or vehicle mass, that can be easily represented as a number;
2. *Qualitative variables*: variables, like policyholder sex or vehicle make, that represent a category and are usually represented with strings.

The quantitative variables, possibly transformed, are already suitable to be used.

To facilitate the use of the qualitative variables, they are usually encoded as sets of binary variables.

If a variable x has only 2 possible modalities, it can be easily encoded in a binary variable x' that assigns 0 to one modality and 1 to the other. For example, if $x =$ sex, it can be encoded this way:

$$x' = \begin{cases} 1 & \text{if sex } = \text{ 'Male'} \\ 0 & \text{if sex } = \text{ 'Female'} \end{cases}$$

In general, if a variable x has K modalities, it can be encoded in $K-1$ binary variables $x'_1, x'_2, \ldots, x'_{K-1}$. For example, if $x =$ make and it can have 4 possible modalities ('Fiat', 'Alfa-Romeo', 'Lancia', 'Ferrari') it can be encoded this way:

$$x'_1 = \begin{cases} 1 & \text{if make } = \text{ 'Fiat'} \\ 0 & \text{otherwise} \end{cases}$$

$$x'_2 = \begin{cases} 1 & \text{if make } = \text{ 'Alfa-Romeo'} \\ 0 & \text{otherwise} \end{cases}$$

$$x'_3 = \begin{cases} 1 & \text{if make } = \text{ 'Lancia'} \\ 0 & \text{otherwise} \end{cases}$$

The variables x'_1, x'_2, x'_3 are called dummy variables. We can observe that all the information about the make is embedded in just these 3 variables, so a fourth dummy variable that indicate the modality 'Ferrari' is not needed. Indeed:

$$\text{make} = \text{'Ferrari'} \iff x'_1 = x'_2 = x'_3 = 0$$

In table 1.1 the dummy variable encoding is illustrated.

Table 1.1: Dummy variables encoding.

Make	x'_1	x'_2	x'_3
Fiat	1	0	0
Alfa-Romeo	0	1	0
Lancia	0	0	1
Ferrari	0	0	0

For some models it is suggested to use also the dummy variable that indicates the K^{th} modality. This encoding is called one-hot encoding and it is mainly used in Neural Networks. For the models considered in this paper the $K-1$ dummy variables encoding is preferred, so we will always consider it.

In the following, when we use the notation $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$, We always consider that the qualitative variables have been already encoded as dummy variables, so $(x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathcal{X} \subseteq \mathbb{R}^p$

1.3.3 Pricing Rule and Modeling

The pricing variables are used as input of a *Pricing Rule*.

Definition 1.5 (Pricing Rule)**.** A *Pricing Rule* is a function $f(\cdot)$ that from an instance of a set of pricing variables $\boldsymbol{x}_i \in \mathcal{X}$ returns a price:

$$\begin{array}{rccc} f: & \mathcal{X} & \longrightarrow & R_+ \\ & \boldsymbol{x}_i & \longmapsto & P_i \end{array}$$

The process of pricing consists in defining a Pricing Rule based on observed data from the past and assumptions on the future.

The first step for defining a Pricing Rule is to model the total cost of claims S and obtain a pricing rule for the risk premium $P^{(risk)}$.

Definition 1.6 (Modeling)**.** Modeling a *response variable* Y means finding a function

$$r : \mathcal{X} \to \mathcal{C}$$

that, given a set of explanatory variables $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathcal{X} \subseteq \mathbb{R}^p$, returns the expected value of the response variable $E(Y)$ and possibly other moments of Y or even the full distribution of Y.

In definition 1.6 we used a generic $\mathcal{C}$ as codomain of the function $r(\cdot)$ to not specify whether the model describes just $E(Y)$ (and so $\mathcal{C} = \mathbb{R}$) or something more, such as the couple $(E(Y), Var(Y))$ or the full distribution of Y.

As we observed in section 1.2.2, under the compound distribution hypotheses, we don't have to model directly the total cost of claims S, but we can separately model N and Z.

1.3.4 Response variables and distributions

Usually in statistical modeling, the response variables are seen as random variables with a distribution belonging to a specified family.

Distribution for the claims count N

The claim count N is a discrete variable with values in $\{0, 1, 2, 3, \ldots\}$. Even if in practice the number of claims can't be arbitrarily high, N is usually modeled with distributions that give a positive probability to all natural numbers. One of the most common distribution used for N is the Poisson distribution.

Definition 1.7 (Poisson Distribution)**.** A random variable N with support $\{0, 1, 2, 3, \ldots\}$ has a Poisson distribution, if its probability function is:

$$p_N(n) = P(N = n) = e^{-\lambda}\frac{\lambda^n}{n!}, \quad \lambda > 0$$

We will indicate it with the notation $N \sim Poisson(\lambda)$.

The Poisson distribution is a parametric distribution that only depends on the parameter λ. In figure 1.2, for different levels of λ the distribution is represented. These plots show how, for larger values of λ, the distribution is shifted to larger values and it is wider.

Indeed, the first two moments are:

$$E(N) = \lambda$$
$$Var(N) = \lambda$$

Thus, increasing λ, both $E(N)$ and $Var(N)$ increase.

Looking to the distribution shape, we can see that:

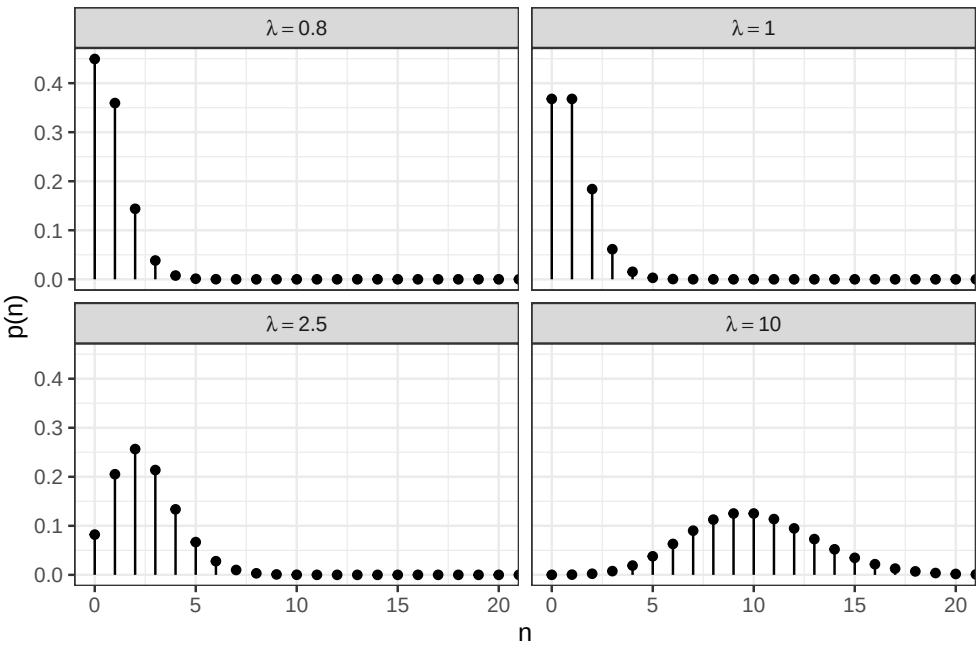


Figure 1.2: Poisson distribution for some values of λ.

- if $\lambda < 1$, the mode is in $n = 0$;
- if $\lambda = 1$, $p(0) = p(1) = \frac{1}{e}$;
- if $\lambda > 1$, the mode is in a value greater than 0 and, as λ increases, the distribution assumes a bell shape similar to the Normal distribution shape. The convergence to the Normal distribution can be obtained with the *Central Limit Theorem*.

In non-life insurance we usually are in the case with $\lambda < 1$. E.g. the average claims frequency for motor third party liability insurances in Italy, in 2018 has been 5.68%[1].

The property $Var(N) = E(N)$ is an important constraint when the distribution is used in practice. It is possible that the observed data shows a different pattern. Often the observed data shows a situation where $Var(N) > E(N)$. This phenomenon is called *overdispersion*.

To address this issue it is possible to use more flexible distributions, such as Negative-Binomial distribution, or to adopt less assumptions on the response variable distribution. One common technique is the Quasi-Poisson model, that we will describe in chapter 2.

Exposure

In section 1.1 we said that non-life insurances usually have a fixed coverage period that usually spans for one year. Often we work with portfolios of insurances with different

[1] ANIA yearly statistical report for motor third party liability

coverage periods. For example, this could be due to the presence of insurances born with shorter coverage periods or to the presence of insurances that has been closed earlier. Moreover, in companies data, often insurance data are collected for accounting years. This means that, if an insurance coverage c spans in two consecutive accounting years a and $a+1$, it is collected as two records: the couple (c, a) and the couple $(c, a+1)$. This situation is quite common, as usually coverages start during the year and not all at the first of the year.

The coverage span for an insurance coverage is called *exposure* and it is usually measured in years-at-risk. For instance, if an insurance coverage spans for 3 months, it corresponds to a quarter of year, so the exposure, measured in years-at-risk, is $v = \frac{1}{4}$. The term year-at-risk comes from the fact that the policyholder exposure is a risk for the insurer, so the exposure is the period in which the insurer is exposed to the risk of paying claims.

It is natural to assume that, if a policyholder has a longer exposure, it is expected for him to experience more claims. Considering that we have to work with policies with different exposures, in order to take this aspect into account, the usual assumption taken is the following: said M the number of claims the policyholder will experience during his period of exposure v (measured in years) and N the number of claims the policyholder would experience during one year, we assume $E(M) = vE(N)$.

This assumption can be further extended if we assume that the claims come from a *Poisson process*.

Definition 1.8 (Counting Process)**.** A stocastic process $\{N(t), t \geq 0\}$ is called *counting process* if:

1. The determination of N(t) are natural numbers
   $N(t) \in \{0, 1, 2, ...\}\ t \geq 0$

2. The process is not decreasing
   $s < t \Rightarrow N(s) \leq N(t)$

In a counting process $\{N(t), t \geq 0\}$:

- $N(t)$ can be interpreted as the number of events or arrivals that occur in the period $[0, t]$;
- $N(t) - N(s),\ s \leq t$ can be interpreted as the number of events or arrivals that occur in the period $]s, t]$. $N(t) - N(s)$ is also called *increment* of the process.

The counting process can be used to model the number of claims that occur to a specific policy.

Definition 1.9 (Poisson Process)**.** A counting process $\{N(t), t \geq 0\}$ is a *Poisson process* with intensity λ if:

1. The increments of the process are stocastically independent
   $\forall n \geq 0, \forall s_1 < t_1 \leq \cdots \leq s_n < t_n$
   $\Rightarrow\ N(t_1) - N(s_1), \ldots, N(t_n) - N(s_n)$ are stocastically independent;
2. The probability of arrival in an interval is proportional to the size of the interval
   $\forall t \geq 0, \forall \Delta t > 0\ \Rightarrow\ P\left(N(t + \Delta t) - N(t) = 1\right) = \lambda \Delta t + o(\Delta t)$
   where $\lim_{\Delta t \to 0} \frac{o(\Delta t)}{\Delta t} = 0$
3. Multiple arrivals are excluded
   $\forall t \geq 0, \forall \Delta t > 0\ \Rightarrow\ P\left(N(t + \Delta t) - N(t) \geq 2\right) = o(\Delta t)$
4. Arrivals at time 0 are almost impossible
   $P\left(N(0) = 0\right) = 1$

Under these hypotheses we obtain the following result:

Theorem 1.1 (Poisson Process)**.** *If* $\{N(t), t \geq 0\}$ *is a Poisson process with intensity* λ*, then:*

$$\forall t \geq 0, \forall \Delta t > 0,\ \Rightarrow\ N(t + \Delta t) - N(t) \sim Poisson(\lambda \Delta t)$$

This result tells us that the distribution of the number of events in any interval $]t, t + \Delta t]$ only depends on the size of the interval Δt. Moreover, for the Poisson property we saw in section 1.3.4, we get:

$$E(N(t + \Delta t) - N(t)) = \lambda \Delta t$$

So, the expected number of arrivals is proportional to the size of the interval Δt. The intensity of the process λ can be also interpreted as the expected number of claims in a unit period.

If we assume that the claims that occur to a policy come from a Poisson process with intensity λ, if we observe that policy for the period $]t, t + v]$, the claims count in that exposure period M are distributed as:

$$M \sim Poisson(v\lambda)$$

In particular, if the observed period spans 1 year, we get:

$$M = N \sim Poisson(\lambda)$$

Distribution for the claim severity Z

The claim severity Z is a continuous variable with values in $[0, +\infty[$. As for the claims count N, even if in practice it can't be arbitrarily high, it is usually modeled with distributions that give a positive density to all the numbers in $]0, +\infty[$. As the null claims are excluded, it is natural to assume $P\left(Z = 0\right) = 0$. One of the most common distribution used for Z is the Gamma distribution.

Definition 1.10 (Gamma Distribution)**.** A random variable Z with support $[0, +\infty[$ has a Gamma distribution, if its probability density function is:

$$f_Z(z) = \frac{\rho^\alpha}{\Gamma(\alpha)} z^{\alpha-1} e^{-\rho z}, \quad \alpha > 0,\ \rho > 0$$

where $\Gamma(\alpha) = \int_0^{+\infty} z^{\alpha-1} e^{-z} \mathrm{d}z$.

We will indicate it with the notation $Z \sim Gamma(\alpha, \rho)$.

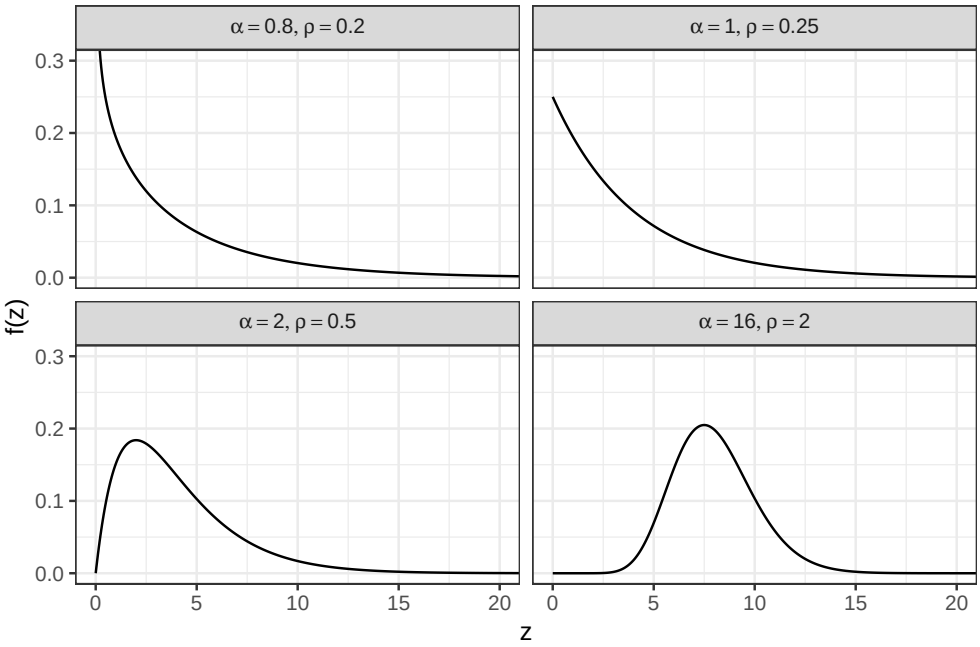


Figure 1.3: Gamma distribution for some values of α and ρ.

The Gamma distribution is a parametric distribution that depends on two parameters:

- $\alpha > 0$, called shape parameter
- $\rho > 0$, called scale parameter

The first two moments of the Gamma distribution are:

$$E(Z) = \frac{\alpha}{\rho}$$
$$Var(Z) = \frac{\alpha}{\rho^2}$$

In figure 1.3, for different levels of α and γ the distribution is represented. These plots show how changing the values of α and γ, the shape changes. We can see that:

- if $\alpha < 1$, $f_z(\cdot)$ is not defined in 0 and it has a vertical asymptote in $z = 0$. In $]0, +\infty]$ it is monotonically decreasing.

- if $\alpha = 1$, $f_z(\cdot)$ starts from $f(0) = \rho$ and then decreases monotonically. In this case, the density function becomes $f_z(z) = \rho e^{-\rho z}$ and the distribution is also called exponential distribution.
- if $\alpha > 0$, $f_z(\cdot)$ starts from $f(0) = 0$, increases until the mode and then decreases.

In figure 1.3 the first three distributions represented have the same expected value $E(Z) = \frac{\alpha}{\rho} = 4$, but different shapes. The third and the fourth have the same variance $Var(Z) = \frac{\alpha}{\rho^2} = 8$, but different expected values. As the shape parameter α increases, the distribution assumes a bell shape similar to the Normal distribution one. The convergence to the Normal distribution can be obtained with the *Central Limit Theorem*.

Another parametrization often used for Gamma distribution is obtained by using the mean μ as a parameter:

$$\mu = \frac{\alpha}{\rho}$$

With this parametrization, the density function becomes:

$$f_Z(z) = \frac{\left(\frac{\alpha}{\mu}\right)^{\alpha}}{\Gamma(\alpha)} z^{\alpha-1} e^{-\frac{\alpha}{\mu}z}, \quad \alpha > 0,\ \rho > 0$$

The advantage of using the parameters (α, μ) is that the link between $E(Z)$ and $Var(Z)$ becomes clearer:

$$E(Z) = \mu$$
$$Var(Z) = \frac{1}{\alpha}\mu^2$$

Computing the coefficient of variation we then obtain:

$$CV(Z) = \frac{\sqrt{Var(Z)}}{E(Z)} = \frac{1}{\sqrt{\alpha}}$$

This result means that the coefficient of variation is constant (given the shape parameter α). As we saw for the Poisson distribution, it is possible that observed data shows a different pattern. In chapter 2, for the Gamma distribution, we will use the parametrization based on (α, μ) instead of the one based on (α, ρ).

Another characteristic of the Gamma distribution that could be problematic in modeling claims severity is that it has a light tail. This means that, as z goes to $+\infty$, $f_Z(z)$ aproaches 0 quite fast. This could lead to a poor fitting for *large claims*. Other distributions with havier tails are for example the *log-Normal* and the *Pareto*.

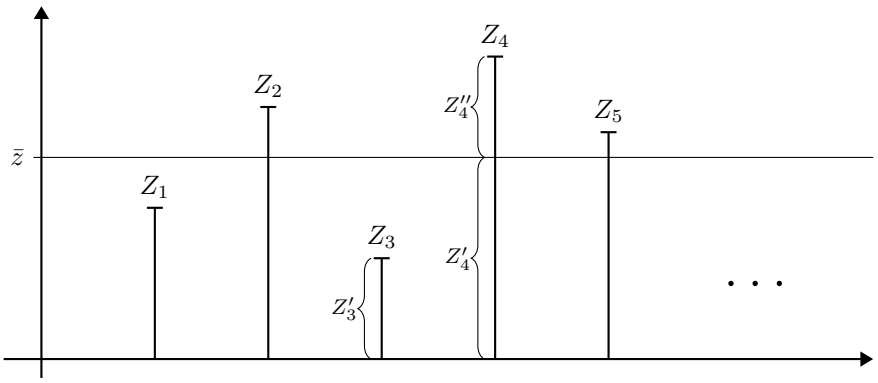


Figure 1.4: Large claims.

Large claims

Modeling large claims in quite difficult in practice because usually there is not a lot of observed data on them, so it is hard to understand if they are related to some risk factors (identifiable by the pricing variables) or they happen just by chance.

First of all, to model large claims, we must define what a large claim is. What is usually done in practice is just choosing a threshold $\bar{z}$ and considering large all the claims with a size that exceeds that threshold. The value $\bar{z}$ must be chosen sufficiently big to consider large the claims above $\bar{z}$, but not so big that there are not enough observed claims that exceeds $\bar{z}$. One common choice for Motor Third Party Liability in European markets could be $\bar{z} = 100'000$€.

Definition 1.11 (Large and Attritional Claims)**.** Given a predetermined threshold $\bar{z}$, we say that:

- a claim Z is a *large claim* if $Z > \bar{z}$
- a claim Z is an *attritional claim* if $Z \leq \bar{z}$

For each claim Z we call:

- *Capped Claim Size*
  $Z' = \min(Z, \bar{z})$;
- *Excess Over the Threshold*
  $Z'' = \max(Z - \bar{z}, 0)$.

In figure 1.4 the *Capped Claim Size* and the *Excess Over the Threshold* are shown. It is easy to show that Z can be decomposed as:

$$Z = Z' + Z''$$

Given the total number of claims N, it can be decomposed as:

$$N = N^{(a)} + N^{(l)}$$

where

- $N^{(a)}$ is the attritional claims count, i.e. the number of claims with size $Z \leq \bar{z}$;
- $N^{(l)}$ is the large claims count, i.e. the number of claims with size $Z > \bar{z}$;

Let's indicate with $Z_{(i)}$ the i^{th} in order from the smallest to the bigger. Sorting the claims we can separate the attritional claims from the large claims as follows:

$$\underbrace{Z_{(1)}, Z_{(2)}, \ldots, Z_{(N^{(a)})}}_{\text{Attritional Claims}}, \underbrace{Z_{(N^{(a)}+1)}, Z_{(N^{(a)}+2)}, \ldots Z_{(N^{(a)}+N^{(l)})}}_{\text{Large Claims}}$$

In order to model the large claims it is possible to use the following three decompositions of the total cost of claims S:

$$\begin{aligned} S &= \underbrace{Z_{(1)} + Z_{(2)} + \cdots + Z_{(N^{(a)})}}_{\text{Attritional Claims}} + \underbrace{Z_{(N^{(a)}+1)} + Z_{(N^{(a)}+2)} + \ldots Z_{(N^{(a)}+N^{(l)})}}_{\text{Large Claims}} \\ &= \underbrace{\sum_{i=1}^{N^{(a)}} Z_{(i)}}_{=S^{(a)}} + \underbrace{\sum_{i=N^{(a)}+1}^{N^{(a)}+N^{(l)}} Z_{(i)}}_{=S^{(l)}} = S^{(a)} + S^{(l)} \end{aligned} \tag{1.11}$$

$$S = \sum_{i=1}^{N} Z_i = \sum_{i=1}^{N} (Z_i I_{Z_i > \bar{z}} + Z_i I_{Z_i \leq \bar{z}}) \tag{1.12}$$

$$S = \sum_{i=1}^{N} Z_i = \sum_{i=1}^{N} (Z'_i + Z''_i) = \sum_{i=1}^{N} (Z'_i + Z''_i I_{Z_i > \bar{z}}) \tag{1.13}$$

These three decompositions of S are useful because they provide three decompositions of $E(S)$:

$$\begin{aligned} E(S) &= E(S^{(a)}) + E(S^{(l)}) \\ &= E(N^{(a)})E(Z|Z \leq \bar{z}) + E(N^{(l)})E(Z|Z > \bar{z}) \end{aligned} \tag{1.14}$$

$$\begin{aligned} E(S) &= E(N)E(Z) \\ &= E(N)\left[P(Z \leq \bar{z})E(Z|Z \leq \bar{z}) + P(Z > \bar{z})E(Z|Z > \bar{z})\right] \\ &= E(N)\left[(1 - P(Z > \bar{z}))\, E(Z|Z \leq \bar{z}) + P(Z > \bar{z})E(Z|Z > \bar{z})\right] \end{aligned} \tag{1.15}$$

$$\begin{aligned} E(S) &= E(N)E(Z) \\ &= E(N)\left[E(Z') + P(Z > \bar{z})E(Z'')\right] \end{aligned} \tag{1.16}$$

1.14, 1.15 and 1.16 provide three approaches to model attritional and large claims:

1. Looking to 1.14 we can model separately attritional claims and large claims. Modeling $N^{(a)}$ and $Z|Z \leq \bar{z}$ we estimate the total cost of claims for the attritional part $S^{(a)}$; modeling $N^{(l)}$ and $Z|Z > \bar{z}$ we estimate the total cost of claims for the large part $S^{(l)}$.
2. Looking to 1.15 we can model together the claim count N, and then we can model the cost of the attritional claims $Z|Z \leq \bar{z}$, the cost of the large claims $Z|Z > \bar{z}$ and the probability to exceed the threshold $P(Z > \bar{z})$.
3. Looking to 1.16 we can model together the claim count N, and then we can model the capped claims size Z', the excess over the threshold Z'' and the probability to exceed the threshold $P(Z > \bar{z})$.

If the large claims component weighs a lot on the total cost of claims, these approaches could lead to quite different estimates of $E(S)$. In particular, if in the observed data the number of large claims is small, it will be hard to model both $N^{(l)}$ and $P(Z > \bar{z})$, so for these components the modeling process could lead to a flat model (i.e. a model without any explanatory variable) or almost flat one (i.e. a model with just few explanatory variables and with mild effects). However, with the first approach, a flat model for $N^{(l)}$ leads to distribute the observed total cost of large claims proportionally to all the policies, while with the second and the third, a flat model for $P(Z > \bar{z})$ leads to distribute the observed total cost of large claims proportionally to the expected number of claims $E(N)$. So, with the first approach, a flat model brings to more solidarity between policies, while, with the second approach, a flat model could bring to an exacerbation of the differences identified by modeling N.

For the second approach we must also introduce a distribution suitable for modeling $P(Z > \bar{z})$.

Binomial distribution

The *binomial distribution* is used to model the counting on events that occurs (successes) in a fixed amount of trials n. For example we can use it to model the number of large claims within a fixed number of n claims.

Definition 1.12 (Binomial Distribution)**.** A random variable Y with support $\{0, 1, 2, \ldots, n\}$ has a Binomial distribution, if its probability function is:

$$p_Y(y) = P(Y = y) = \binom{n}{y} p^y (1-p)^{n-y}, \quad p \in [0, 1]$$

We will indicate it with the notation $Y \sim Binom(n, p)$.

The binomial distribution is a parametric distribution that depends on the parameters n and p. n represents the number of trials, while p represents the probability for a trial to succed. The assumption is that the n trials are identical, so they have all

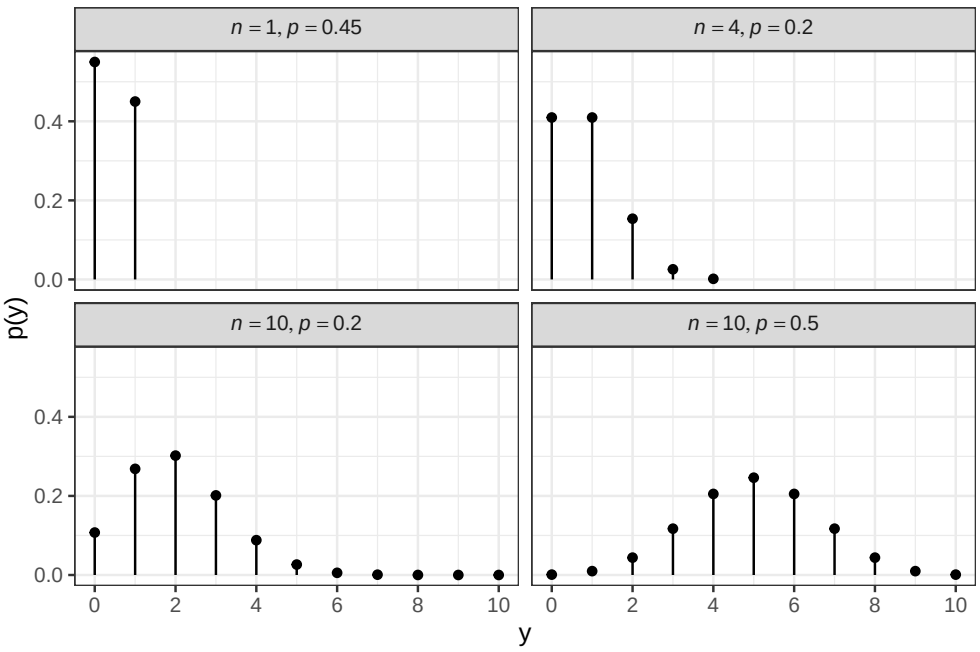


Figure 1.5: Binomial distribution for some values of n and p.

the same probability p to succed. In figure 1.5 the distribution is represented for different levels of n and p.

The first two moments of the binomial distribution are:

$$E(N) = np$$
$$Var(N) = np(1-p)$$

If $n = 1$, the binomial distribution assumes only the values 1 (with probability p) and 0 (with probability $1 - p$). In this case it is also called *Bernoullian distribution* and it can be used to model the indicator of an event I_E.

If $n > 1$, the binomial distribution assumes a shape centered on its expected value $E(Y) = np$ and fading for values of y that moves away from $E(Y)$. As n increases, the distribution assumes a bell shape similar to the Normal distribution shape. The convergence to the Normal distribution can be obtained with the *Central Limit Theorem*.

From the binomial Distribution it is also possible to define the scaled binomial distribution by dividing its value by n.

Definition 1.13 (Scaled Binomial Distribution)**.** If $Y \sim Binom(n, p)$, and $Y' = \frac{Y}{n}$, we will say that Y' has a *Scaled Binomial Distribution* and we will indicate it with the notation $Y' \sim Binom(n, p)/n$.

The support of Y' is $\{0, \frac{1}{n}, \frac{2}{n}, \ldots, 1\}$ and its probability function is:

$$p_{Y'}(y') = P\left(Y' = y'\right) = \binom{n}{ny'} p^{ny'}(1-p)^{n-ny'}, \quad p \in [0,1]$$

In chapter 2 we will use the Scaled Binomial Distribution.

In non-life insurance pricing, the binomial distribution can be used to model the probability for a claim to have specific characteristics. For example we can use it to model the probability that a certain claim is a large one, $P(Z > \bar{z})$, in order to model separately attritional claims severity $\{Z|Z \leq \bar{z}\}$ and large claims severity $\{Z|Z > \bar{z}\}$, as we have seen in section 1.3.4.

Another example is the decomposition between claims with only material damages and claims with also bodily injuries. Modeling separately these two components is useful because they usually have a different distribution for the claim size.

As for large claims we can decompose S in the following two ways:

$$\begin{aligned} E(S) &= E(S^{(\text{things})}) + E(S^{(\text{inj})}) \\ &= E(N^{(\text{things})})E(Z|\bar{J}) + E(N^{(\text{inj})})E(Z|J) \end{aligned} \tag{1.17}$$

$$\begin{aligned} E(S) &= E(N)E(Z) \\ &= E(N)\left[P(\bar{J})E(Z|\bar{J}) + P(J)E(Z|J)\right] \\ &= E(N)\left[(1-P(J))\,E(Z|\bar{J}) + P(J)E(Z|J)\right] \end{aligned} \tag{1.18}$$

where:

- $N^{(\text{things})}$ is the number of claims with only material damages;
- $N^{(\text{inj})}$ is the number of claims with injuries;
- J is the event that represents that a specific claim presents injuries; such as Z is a representative for $Z_1, Z_2, \ldots, Z_N$, J is a representative for $J_1, J_2, \ldots, J_N$.

Combining this decomposition with what we have seen in large claims decomposition, we can further develop our decomposition taking into account both the presence or absence of injuries and the occurrence or not of a large claim. One example could be:

$$\begin{aligned} E(S) &= E(N)\left[(1-P(J))\,E(Z|\bar{J}) + P(J)E(Z|J)\right] \\ &= E(N)\,\{ \\ &\quad (1-P(J))\,E(Z|\bar{J}) \\ &\quad + P(J)\,[\\ &\qquad P(Z \leq \bar{z}|J)E\left(Z \mid Z \leq \bar{z} \wedge J\right) \\ &\qquad + P(Z < \bar{z}|J)E\left(Z \mid Z < \bar{z} \wedge J\right) \\ &\quad] \\ &\} \end{aligned}$$

This way, we are decomposing only the claims with injuries between attritional and large. That makes sense because claims that don't produce injuries usually have small severities.

1.3.5 Model fitting and data available

Once we have chosen how to decompose S, we have to model the response variables needed for that decomposition (N, Z, I_J, ...) with the explanatory variables. Thus we have to estimate a function $r : \mathcal{X} \to \mathcal{C}$ as defined in 1.6.

In order to estimate $r(\cdot)$ we have also to take some assumptions on the distribution of the response variable and on the shape of $r(\cdot)$. We will call *model* a set of assumptions on the response variable and on the shape of $r(\cdot)$. We will discuss some of the most widespread models for claims count and claims severity in chapter 2.

Defined the model, we have to estimate it using observed data. In general, to model a response variable Y_i with the explanatory variables $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip}) \in \mathcal{X} \subseteq \mathbb{R}^p$, the observed data is in the form:

$$\mathcal{D} = \{(\boldsymbol{x}_1, w_1, y_1),\ (\boldsymbol{x}_2, w_2, y_2),\ \ldots,\ (\boldsymbol{x}_i, w_i, y_i),\ \ldots,\ (\boldsymbol{x}_n, w_n, y_n)\}$$

where:

- n is the number of observations in the dataset;
- $\boldsymbol{x}_i \in \mathcal{X} \subseteq \mathbb{R}^p$ is the set of explanatory variable for the observation i;
- w_i is the weight for the observation i;
- $y_i \in \mathcal{Y} \subseteq \mathbb{R}$ is the realization of the response variable Y_i for the observation i.

What an observation is, depends on the variable we are modeling. For instance:

- If we are modeling the yearly claim count N_i, each observation could be a policy (or a couple (policy, accounting year)), the weights could be the exposures v_i and the realizations of response variables could be the number of observed claims for that policy (or couple (policy, accounting year)).
- If we are modeling the claim severity Z_j, each observation could be a claim j, the weights could all be 1 and the realizations of the response variable could be the observed cost for the claim j. It is also possible to model the claim severity taking into account the total cost of claims for the policy $S_i = \sum_{j=1}^{N_i} Z_j$. In this case, each observation would be a policy i, the weights would be the number of claims for each policy n_i and the realizations of response variables would be the total observed cost for the claims of the policy i.
- If we are modeling the occurrence of injuries in a claim I_{Jj}, each observation could be a claim j, the weights could be all 1 and the realizations of response variables could be an indicator that assume the value 1 if the claim j caused injuries and 0 otherwise. As for the claim severity, we can also aggregate data for policy, so each observation would be a policy i, the weights would be the number of claims n_i

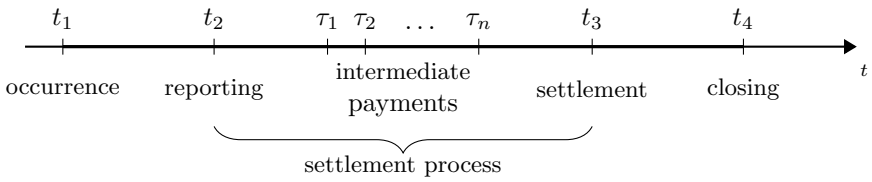


Figure 1.6: Claim timeline.

for the policy i and the realizations of response variables would be the number of claims that caused injuries among the claims of the policy i.

In each of these cases, y_i is seen as a realization of the random variable Y_i. With an inferential process we obtain estimations on Y_i distribution based on observations of their realizations y_i.

Settlement process and IBNR claims

One of the challenges in non-life insurance pricing is that obtaining the observed data is not so straightforward. In many insurance coverages, such as MTPL, the settlement process could last many years, so, if we want to develop models using data from recent years, not all the information is available. To better understand this aspect we have to discuss how the settlement process works.

In figure 1.6 the settlement process for a claim is represented. At time t_1 the insured event (e.g. an accident) occurs. From this moment a liability for the insurer emerges, even if the insurer has not been notified yet. This liability is called *Outstanding Loss Liability*. In t_2 the claim is reported and the insurance is notified about the occurrence of the event. From this moment the settlement process starts. This process consists in evaluating the event and understanding the responsibilities of the parts and the entity of the damage. During this process, controversies between the parts can emerge and, in particular if injuries occurred, the damage evaluation can take a lot of time. When the situation is clear and everything is defined, the claim is settled and the liabilities are paid. In t_3 we have the settlement and in t_4 the claim is closed. In is possible that $t_4 = t_3$, but it is not always the case. If the settlement process takes a long time and the insurer already knows he will have to pay something, he can make some partial payments during the period $[t_2, t_3]$. These intermediate payments are paid at times $\tau_1, \tau_2, \ldots, \tau_n \in [t_2, t_3]$. It is also possible that a claim is opened and then gets closed without any payment. After the closing (t_4) it is also possible that a claim is reopened and that more payments emerge.

From the moment the claim is reported (t_2), the insurer estimates how much he is going to pay for that claim and he allocates that sum in a reserve. As new information emerges and some payments are settled, the reserve is updated. The aim for this reserve is to have a best estimate for the future payments for the claims already emerged. As the claim gets settled, the sum between the paid and the reserved converges to the

final cost of the claim.

From this description emerges that:

- In the period $]t_1, t_2[$ the insurer has an outstanding loss liability for an event that has not been reported yet; in this case we will talk about Incurred But Not yet Reported claim (IBNyR).
- In the period $[t_2, t_3[$ the insurer has an outstanding loss liability for an event that has been reported, but has not been totally settled yet, so that liability is just an estimate; in this case we will talk about Incurred But Not enough Reported claim (IBNeR).

Model fitting with available data

The IBNyR and IBNeR issue is particularly challenging when we have to perform a risk evaluation at a specific time t. In general $t_1, t_2, \ldots$ are not known a priori, so we don't know if in the future more claims for accidents occurred in the past will be reported and we don't know if the ones that are already reported will experience a revaluation. That means that, in general, when we model N and Z at a specific time t, we can't observe the total number of claims occurred to each policy n_i and the payments for each claim z_j. What we can use is:

- $n_i^{(t)} = n_i^{(\text{reported in } t)}$
  where:
  - $n_i^{(\text{reported in } t)}$ is the number of reported claims in t for the policy i;
- $z_j^{(t)} = z_j^{(\text{paid in } t)} + z_j^{(\text{reserved in } t)}$
  where:
  - $z_j^{(\text{paid in } t)}$ is the amount already paid in t for the claim j;
  - $z_j^{(\text{reserved in } t)}$ is the amount reserved in t for the claim j.

When we use this data for modeling the total cost of claims we must be particularly aware on what we are using. In general:

$$n_i^{(t)} \neq n_i, \qquad z_j^{(t)} \neq z_j$$

The common case is that $n_i^{(t)} < n_i$ and $z_j^{(t)} < z_j$. If we used $n_i^{(t)}$ and $z_j^{(t)}$ without any correction, we would underestimate both $E(N)$ and $E(Z)$, obtaining a biased estimate for $E(S)$.

To tackle these problems what is usually done is fitting the models for S_i with $n_i^{(t)}$ and $z_j^{(t)}$ and then apply a flat corrective coefficient α to $\widehat{E(S_i)}$ based on an aggregated estimate of $E(S)$ that takes into account the long settlement process.

An estimate for the expected total cost of claims for a generic policy in the portfolio $E(S)$ can be obtained with techniques based on runoff triangles, such as the *Chain*

Ladder. These techniques are based on projecting the cost of claims already emerged to the final total cost of claims. We are not going to discuss these techniques in this thesis. We just have to know that these techniques provide us with an estimate for $E(S)$. Let's call it $\widehat{E(S)}^{CL}$. This estimate does not depend on explanatory variables; it is a sort of average total cost of claims for the policies in the portfolio.

Meanwhile, with the available data $n_i^{(t)}$ and $z_j^{(t)}$, the fitting for all the models needed in the decomposition of S is performed and, for each policy $i \in \{1, 2, \dots, n\}$, $E(S_i)$ is obtained. Let's call it $\widehat{E(S_i)}'$. As we used the data available in t that comes from claims not totally settled, $\widehat{E(S_i)}'$ is a biased estimate for $E(S_i)$.

We can then balance the estimates $\widehat{E(S_i)}'$ with $\widehat{E(S)}^{CL}$ by computing:

$$\alpha = \frac{n}{\sum_{i=1}^{n} \widehat{E(S_i)}'} \widehat{E(S)}^{CL}$$

and applying to the estimates as follows:

$$\widehat{E(S_i)} = \alpha \, \widehat{E(S_i)}'$$

We will call $\widehat{E(S_i)}$ rebalanced estimates.

The property of these rebalanced estimates is that on average they are equal to $\widehat{E(S)}^{CL}$:

$$\begin{aligned}
\frac{\sum_{i=1}^{n} \widehat{E(S_i)}}{n} &= \frac{\sum_{i=1}^{n} \alpha \widehat{E(S_i)}'}{n} \\
&= \alpha \frac{\sum_{i=1}^{n} \widehat{E(S_i)}'}{n} \\
&= \frac{n}{\sum_{i=1}^{n} \widehat{E(S_i)}'} \widehat{E(S)}^{CL} \frac{\sum_{i=1}^{n} \widehat{E(S_i)}'}{n} \\
&= \widehat{E(S)}^{CL}
\end{aligned}$$

So, if $\widehat{E(S)}^{CL}$ is a unbiased estimator for $E(S)$, we obtain:

$$E\left(\frac{\sum_{i=1}^{n} \widehat{E(S_i)}}{n}\right) = E\left(\widehat{E(S)}^{CL}\right) = E(S)$$

This procedure can be further developed by balancing not directly the total cost of claims $E(S)$, but its components. For example, we could separately balance the total cost of claims that only caused damage to things and the total cost of claims that

caused injuries. This separation in components can lead to a more precise estimate because usually claims that caused injuries have a slower settlement process so they will have a higher corrective coefficient α.

If the dataset contains policies from many years and during the last years a relevant change in the portfolio risk mixture happened, it is also possible to compute α only with the policies from the last year of the dataset, rather than with all the n policies of the dataset.

The fact that the final estimates $\widehat{E(S_i)}$ are rebalanced on $\widehat{E(S)}^{CL}$ means that the explanatory variables effects estimated with $n_i^{(t)}$ and $z_j^{(t)}$ are used just as relative effects and not absolute ones. For instance, if the model says that young people have an expected total cost of claims $\widehat{E(S_i)}'$ that is two times the old people one, that relative coefficient 2 will be kept also in the balanced estimate $\widehat{E(S_i)}$.

For this reason, in practice, often the modeling is considered composed in 2 parts:

1. *Tariff Requirement* (or *Fabbisogno Tariffario*): the estimate of $\widehat{E(S)}^{CL}$ by aggregated data;
2. *Personalization*: the estimate of $\widehat{E(S_i)}'$ and the relative coefficients.

The techniques used for *Tariff Requirement* are employed also to estimating the Claim Reserve, that is a fundamental component of the financial statement in Non-Life insurace companies.

1.4 Beyond technical pricing

In section 1.2.3 we defined:

- the *Risk Premium*
  $P^{(risk)} = E(S)$
- the *Technical Price*
  $P^{(tech)} = E(S) + E$

In section 1.3 we described how the risk premium can be estimated. In this thesis we are not going to deal with the estimate of the expenses.

In this section we are going to discuss what the *Tariff* and the *Offer Price* are and which are the further needs that the offer should satisfy. The following description is referred to MTPL insurance in the Italian market. Most of the comments we make can be applied to other motor coverages too.

1.4.1 Tariff and Offer Price

The *Tariff* is the official price for the policy. Over the cost of claims and the expenses, it must include all the loadings for cost of capital and profits. The tariff has a particular importance because it is subjected to strict regulations and it must be approved by the supervisory authority, that in Italy is the IVASS (Istituto per la vigilanza sulle Assicurazioni).

In section 1.3.1 we described some of the explanatory variables that can be used to build the technical price. For technical pricing there are no constraints because it is used only for internal monitoring and the final price proposed to the client does not directly depend on it. However, some of the variables used for technical pricing can't be used in tariff. In particular, the regulations dictete that companies can't discriminate clients based on sex, ethnic group, religion or place of birth. Thus, for example, even if from statistical data we see that women usually experience less claims than men, we can't discriminate men by offering them a higher price. Moreover, some variables have constraints on tariff coefficients. For example, in MTPL insurance, the bonus-malus class is strongly regulated. Every company must recognize the bonus-malus class matured by clients (even if they matured them with other companies) and the coefficients of this variable must be monotonically increasing, i.e. a lower class must correspond to a better tariff (in the Italian bonus-malus system the lower the class the better the premium). Another tariff constraint is that for some coverage, such as MTPL, the insurer has an obligation to contract. That means that whoever the client is, independently of how risky he is, the company must offer a premium and, if the client accepts, the company must underwrite the insurance contract. In this context, if the company offers an unreasonably high premium, it could fall in an attempt to avoiding the obligation to contract. For this reason, the tariff can't be arbitrarily high and must contemplate a maximum premium. To be sure that all the costraints has been respected, the tariff, before entering in production, must follow a strict approval process.

To make the offer price more flexible and to facilitate business competition, the supervisory authority allows insurance companies to sell policies not at the tariff price, but at the price obtained subtracting from it a discount $D_i \geq 0$. The premium obtained this way is called *Offer Price*.

$$P_i^{(\text{offer})} = P_i^{(\text{tariff})} - D_i$$

That means that, for the offer price to adequately cover the cost of claims and expenses, the tariff must include a loading for discounting. This loading for discounting, called *discounting flexibility*, can be partially spent by the agent and partially by the insurance company itself. The discounts can change over time in a much more agile way than the tariff. For example in Italy, during the Spring 2020 Covid19 crisis, many companies introduced measures to support customers needs with important discounts on both new businesses and renewals. From a technical point of view, these discounts have been funded by the remarkable decrease on claim frequency due to the reduced

traffic. Discount measures like these are welcomed by the supervisory authority because they promote business competition and lead to lower prices for consumers.

1.4.2 Price Optimization

Both tariff and offer price must be based not only on technical logic, but also on commercial logic. They are determined with a process of *Price Optimization*. The final goal for a company is to maximize profits this year and in the next ones, so the objective of price optimization must be obtaining the optimal price to reach this goal. Maximizing profits is a quite generic goal and can't be easily expressed as an analytical optimization problem. For this reason the pricing choices can be guided by the business strategy that can be translated in specific *Key Performance Indicators* (KPI) that have to be optimized. In this optimization framework, the technical price can be seen as an estimate of the expected cost related to the policy. Knowing the costs it is possible to tune the final premium by working on margins.

The components that act on price optimization can be addressed to:

1. *technical pricing*;
2. *client expectation*;
3. *business strategy*.

We already extensively covered technical pricing in previous chapters.

Client expectation is basically the price that the client is willing to pay for the specific product. If the client would pay a premium higher than the technical one, the insurance company has the space for determining an offer price higher than the technical one and gaining margins on that contract. To analyze client expectation, what is usually done is:

- for new business modeling his conversion probability;
- for renewal business modeling his retention probability.

For example some guarantees or some options are perceived by the clients as being really worth even if their technical price is not so high. The perception of the client depends also on the competitors pricing and how easy comparing offers from different companies is. In the last years, in the Italian market, the development of aggregators has made much easier for consumers to compare offers from different companies, increasing the competition and the attention on pricing. Anyway, if a company is able to differentiate itself from the others and to make its product be perceived as more valuable, it can sell it at a higher price than other companies. For example this can be achived by improving customer care and customer experience.

If the technical price and the conversion probability functions are given, finding the optimal price for a policy can be expressed as an analytical optimization problem. However, to find the optimal price, one should also take into account that usually policies are not sold alone, but in packages of guarantees. With a wider vision, a business strategy

could be selling MTPL policy with almost no margins if it allows to sell other guarantees with higher margins. Moreover, as the aim is not to be profitable this year, but also in the following ones, the company should also consider the *lifetime value* of the client. Indeed, a satisfied client will also stipulate other contracts in the future and can bring to the company other clients from his connections. So, selling a policy with small margins today can lead to high margins tomorrow in other policies.

The business strategy could also contemplate being more aggressive on certain targets of client and less on others. For example, if the company is particularly strong in certain regions, it could make sense for it to push in that region to further increase its market share. Vice versa, in regions where the company doesn't sell much, it could be safer not to push too much and to be more careful. In a risk management framework, this can be also interpreted as introducing a further risk margin for clusters where there isn't enough observed data and the lack of information brings to more uncertainty. An aggressive pricing can also make sense for a young company that is growing and it is not supposed to be profitable from the first years. From a marketing point of view this strategy can increase the brand awareness by the clients and can strengthen the company image.

Anyway, a company can't arbitrarily discount policies because an excess in discounting could cause severe drawbacks on a financial perspective. Therefore, an insurance company must always respect the solvency constraints defined by the supervisory authority to safeguard itself from bankruptcy. The company solvency is essential to protect all the stakeholders, that are both the clients and the investors.

1.5 The actuary role

In this technical pricing and price optimization framework, the actuary is the one that conducts the analysis and defines the pricing rules. The *International Actuarial Association* (IAA) describes the actuaries as "highly qualified professionals who analyze the financial impact of risk for organizations like insurers, pensions fund managers, and more" and it states that their work "requires a combination of strong analytical skills, business knowledge, and understanding of human behavior".[2]

First of all, the actuary must master the main statistical and data science techniques used to develop models for technical pricing. On this field, in the last years, the development of machine learning and high performance computing has permitted a huge development of technical pricing allowing actuaries to use much more complex variables and models. However, the actuary does not have just to be an expert in statistics and machine learning. He must be also able to interpret the results he gets with his models and use his expertise to understand if the results he gets are fine for future predictions. As we already mentioned, the pricing rules will be used for policies that will be sold in the future, so they have to be defined with a mixture of observation of

[2]IAA, About Actuaries

the past and assumptions on the future. In addition, sometimes it is needed to define prices for clusters where the company have no historical data. This can happen when a company is expanding to new customers for example by opening new selling channels or by pushing in regions where its market share is quite small. Furthermore, the lack of historical data can be due to the full sector evolution. For example, in these years, new vehicles, such as electric cars and cars with Advanced Driver-Assistance Systems (ADAS), are spreading. As these vehicles didn't exist in the past, historical data doesn't exist. So, finding the proper pricing is challenging. From the company point of view, positioning with a competitive pricing on these segments is important for future business, but the risk must be properly evaluated. For these kind of tasks, the actuary must have a deep domain knowledge on the field.

In the last years, the increase of competition brought to an increase in price optimization importance. Now most of the companies build their own conversion and retention probability models and they are developing more complex business strategies. In this context, it is fundamental for the actuary to understand the clients behaviors, in order to optimize tariff and offer price.

The importance for price optimization implies that the technical pricing must not be conducted independently from tariff and offer pricing. Even in companies where the technical pricing and the offer pricing are carried out by two separate teams, the two teams have to collaborate and coordinate together. This need has some relevant implications on how technical pricing is conducted that we will further discuss in section 2.2.3.

CHAPTER 2

Statistical models for Non Life Insurance Pricing

In this chapter we are going to illustrate some of the most widespread models for technical pricing. For each model we are going to describe its benefits and drawbacks and in section 2.2 we will compare them by discussing how they fit the pricing needs.

2.1 Statistical Models

In this section we will start by describing the Generalized Linear Model (GLM), that is the most employed model in technical pricing, to then present some of its advancements: the Generalized Additive Model (GAM), the Shrinkage estimators for GLM and the Bayesian GLM. After this description, we will also present the Gradient Boosting Machine (GBM), that is one of the most effective general purpose machine learning models. This allows us to have a comparison between GLM based models and general purpose machine learning models.

2.1.1 Generalized Linear Models

Linear Exponential Families

One of the GLM assumptions is that the response variables belong to a *Linear Exponential Family*. In this section we are going to explain what a linear exponential family is and which distributions fit its definition.

Definition 2.1 (Linear Exponential Family)**.** A Linear Exponential Family $\mathcal{F}$ is a parametrical family of probability distributions with density function (or probability function in the discrete case) that can be expressed in the form:

$$f(y;\theta,\lambda) = \exp\left\{\frac{y\theta - b(\theta)}{\lambda}\right\}c(y,\lambda), \quad y \in \mathcal{Y} \subseteq \mathbb{R}$$

where:

- $\theta \in \Theta \subseteq \mathbb{R}$ is called *canonical parameter*;
- $\lambda \in \Lambda \subseteq]0,+\infty[$ is called *dispersion parameter*;
- $b : \Theta \to \mathbb{R}$ is a real function called *cumulant function*;
- $c : (\mathcal{Y}, \Lambda) \to [0,+\infty[$ is a real function;
- Θ is a non degenerate interval, i.e. intΘ is not empty.

An exponential family $\mathcal{F}$ is characterized by the elements $(\Theta, b(\cdot), \Lambda, c(\cdot,\cdot))$. By properly choosing the sets Θ, Λ and the functions $b(\cdot), c(\cdot,\cdot)$, it is possible to obtain many useful families.

It can be easily shown that the families Normal, Poisson, Gamma and Binomial are exponential families. In table 2.1 the characterizations for these exponential families are reported.

Table 2.1: Some Linear Exponential Families.

Distribution	**Notation**	Θ	θ	Λ	λ	$b(\theta)$
Normal	$N(\mu,\sigma^2)$, $\mu \in \mathbb{R}$ $\sigma \in]0,+\infty[$	$\mathbb{R}$	μ	$]0,+\infty[$	σ^2	$\frac{\theta^2}{2}$
Poisson	$Poisson(\mu)$, $\mu \in]0,+\infty[$	$\mathbb{R}$	$\log(\mu)$	$\{1\}$	1	e^θ
Gamma	$Gamma(\alpha,\mu)$, $\alpha \in]0,+\infty[$ $\mu \in]0,+\infty[$	$]-\infty,0[$	$-\frac{1}{\mu}$	$]0,+\infty[$	$\frac{1}{\alpha}$	$-\log(-\theta)$
Scaled Binomial	$Binom(n,p)/n$, $n \in \mathbb{N}$ $p \in]0,1[$	$\mathbb{R}$	$\log\left(\frac{p}{1-p}\right)$	$\left\{\frac{1}{n}\right\}$	$\frac{1}{n}$	$\log\left(1+e^\theta\right)$

The distributions that belong to an exponential family have many useful properties. For example they are provided with all the moments and their moments can be obtained using the derivatives of the cumulative function $b(\cdot)$. If Y is a random variable with distribution belonging to an exponential family $\mathcal{F}$ with parameters θ, λ, its first two moments are:

$$E(Y) = b'(\theta) \tag{2.1}$$
$$Var(Y) = \lambda b''(\theta) \tag{2.2}$$

As, within a specified family, the parameters θ and λ determine a distribution, in practical problems the object of estimation will be the couple (θ, λ). In many problems it is natural to consider distributions from a linear exponential family where the dispersion parameter can be expressed as $\lambda = \frac{\phi}{\omega}$, where $\omega > 0$ is a known *weight* and $\phi > 0$ is a parameter that we will keep calling *dispersion parameter*. In this case, the density of probability function depends on the parameters θ and ϕ and will be expressed as:

$$f(y; \theta, \phi, \omega) = \exp\left\{\frac{\omega}{\phi}\left[y\theta - b(\theta)\right]\right\} c(y, \phi, \omega), \quad y \in \mathcal{Y} \subseteq \mathbb{R}$$

In this case the parameters θ and ϕ will be object of estimation, while ω is an already known value. As we will see later, this representation allows us to consider as known weights:

- the exposure v in the Poisson distribution;
- the number of trials n in the Binomial distribution.

Model assumptions

Let's assume that, for n statistical units, the observations $\mathcal{D} = \{(\boldsymbol{x}_1, \omega_1, y_1), \ldots, (\boldsymbol{x}_n, \omega_n, y_n)\}$ are available, where $\boldsymbol{x}_i$ is a vector of explanatory variables determinations, ω_i is a known weight and y_i is the response variable determination. $\boldsymbol{x}_i, \omega_i, y_i$ are all real numbers. The vector $\boldsymbol{y} = (y_1, \ldots, y_n)^t$ is considered a determination of the response random vector $\boldsymbol{Y} = (Y_1, \ldots, Y_n)^t$.

In GLM we assume that:

1. The response variables $Y_1, \ldots, Y_n$ are stochastically independent and with probability distribution belonging to a same linear exponential family; i.e. the probability distribution of Y_i has density function (or probability function in the discrete case) that can be expressed as:

$$f(y_i; \theta_i, \phi, \omega_i) = \exp\left\{\frac{\omega_i}{\phi}\left[y_i\theta_i - b(\theta_i)\right]\right\} c(y_i, \phi, \omega_i), \quad y_i \in \mathcal{Y} \subseteq \mathbb{R}$$

   We highlight that only θ_i and ω_i depend on i, while the dispersion parameter ϕ is the same for all the observations.

2. The explanatory variables determinations vector $\boldsymbol{x}_i = (1, x_{i1}, \ldots, x_{ip})^t$ affects the probability distribution of the response variable Y_i by the linear predictor:

$$\eta_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip}$$

that is a linear function of the regression parameters $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)$.

3. The linear predictor η_i is linked to the expected value of the response variable $\mu_i = E(Y_i)$ by the following relation:

$$g(\mu_i) = \eta_i = \boldsymbol{x}_i^t \boldsymbol{\beta}$$

where $g : \mathbb{R} \to \mathbb{R}$ is a monotonic function with continuous first and second derivatives. $g(\cdot)$ is called *link function*.

Often, the assumption 1 is called stochastic assumption, while the 2 and 3 are called structural assumptions.

Let's indicate with $\boldsymbol{X}$ the design matrix, i.e. the matrix in which each row $\boldsymbol{x}_{i\cdot}$ represents the vector of the explanatory variables for the observation i and each column $\boldsymbol{x}_{\cdot j}$ represents the vector of the observations for the explanatory variable j. The design matrix is represented in figure 2.1. The matrix starts with a column of 1s, that is used to model the intercept. Thus, it is a matrix $n \times (p+1)$. We assume, as it is common in actuarial datasets, that $n > p + 1$.

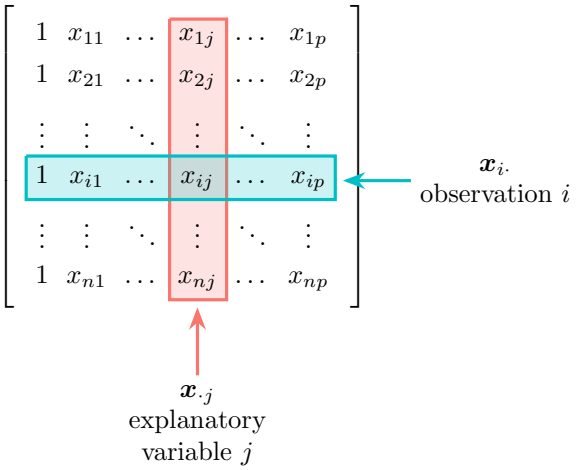


Figure 2.1: Design Matrix $\boldsymbol{X}$.

We can then express the GLM structural assumptions in a matrix form as:

$$\boldsymbol{g}(\boldsymbol{\mu}) = \boldsymbol{X}\boldsymbol{\beta}$$

where $\boldsymbol{g}(\cdot)$ must be intended as the vectorial function that links every μ_i to $g(\mu_i)$.

$$\begin{array}{rccc} \boldsymbol{g}: & \mathbb{R}^n & \longrightarrow & \mathbb{R}^n \\ & \begin{pmatrix} \mu_1 \\ \vdots \\ \mu_n \end{pmatrix} & \longmapsto & \begin{pmatrix} g(\mu_1) \\ \vdots \\ g(\mu_n) \end{pmatrix} \end{array}$$

We assume the design matrix to be a full rank matrix, i.e. rank$(\boldsymbol{X}) = p + 1$. This assumption corresponds to assuming that the columns of $\boldsymbol{X}$ are linearly independent.

The function $g(\cdot)$ can be chosen as any monotonic function with continuous first and second derivatives. Given a family $\mathcal{F}$, a common choice is its canonical link function that is defined as:

$$g(\mu) = b'^{-1}(\mu)$$

From (2.1) we obtain that, as $\mu = b'(\theta)$, choosing the canonical function corresponds to using θ as the linear predictor:

$$\eta = g(\mu) = b'^{-1}(\mu) = \theta$$

In table 2.2 the canonical link functions for the families mentioned in 2.1 are reported.

Table 2.2: Canonical link functions.

Distribution	**Cumulant function** $b(\theta)$	**Derivative** $b'(\theta)$	**Canonical link function** $g(\mu) = b'^{-1}(\mu)$
Normal	$\frac{\theta^2}{2}$	θ	μ
Poisson	e^θ	e^θ	$\log(\mu)$
Gamma	$-\log(-\theta)$	$-\frac{1}{\theta}$	$-\frac{1}{\mu}$
Scaled Binomial	$\log\left(1+e^\theta\right)$	$\frac{e^\theta}{1+e^\theta}$	$\log\left(\frac{p}{1-p}\right)$

In the Gamma case, its canonical function $g(\mu) = -\frac{1}{\mu}$ has the drawback that it links the expected values $\mu \in]0, +\infty[$ to $\eta \in]-\infty, 0[$. This would require some constraints on $\boldsymbol{\beta}$ because $\eta = \boldsymbol{x}^t\boldsymbol{\beta}$ would have to be < 0. For this reason, it is preferred to use $g(\mu) = \log(\mu)$ that maps $]0, +\infty[$ to $\mathbb{R}$.

In the Scaled Binomial case the canonical function $g(p) = \log\left(\frac{p}{1-p}\right)$ is called logit and its inverse $g^{-1}(\eta) = \frac{e^\eta}{1+e^\eta}$ is called logistic. For Scaled Binomial distribution we keep using the notation p for the expected value as it corresponds to the probability of success p.

Model fitting

The model depends on the parameters $(\boldsymbol{\beta}, \phi)$. Indeed, the parameters θ_i can be obtained by $\boldsymbol{\beta}$ as:

$$\theta_i = b'^{-1}(\mu_i) = b'^{-1}(g^{-1}(\eta_i)) = b'^{-1}\left(g^{-1}\left(\boldsymbol{x}_i^t\boldsymbol{\beta}\right)\right)$$

Therefore, fitting the model corresponds to estimating $(\boldsymbol{\beta}, \phi)$. The technique used in GLM is the *Maximum Likelihood*. Let's indicate with $L(\boldsymbol{\beta}, \phi; \boldsymbol{y})$ the model likelihood. We remind that the likelihood is a function of the parameters that maps $(\boldsymbol{\beta}, \phi)$ to the density (or probability in the discrete case) of the observed values $\boldsymbol{y}$ conditioned to the parameters $(\boldsymbol{\beta}, \phi)$

$$\begin{array}{rccc} L: & \mathbb{R}^{p+1} \times \Lambda & \longrightarrow & [0, +\infty[\\ & (\boldsymbol{\beta}, \phi) & \longmapsto & f_{\boldsymbol{Y}}(\boldsymbol{y}; \boldsymbol{\theta}, \phi) \end{array}$$

The maximum likelihood estimates are the values $(\boldsymbol{\beta}, \phi)$ that maximize $L(\boldsymbol{\beta}, \phi; \boldsymbol{y})$. In practice, $\boldsymbol{\beta}$ are the parameters of interest, while ϕ is considered as a disturbance parameter. It is possible to show that conditioned to any ϕ, the value for $\boldsymbol{\beta}$ that maximizes $L(\cdot, \cdot)$ does not depend on ϕ. Therefore, $\boldsymbol{\beta}$ and ϕ can be estimated separately.

Let's indicate with $\tilde{\boldsymbol{\beta}}$ the maximum likelihood estimator for $\boldsymbol{\beta}$. Its determination $\hat{\boldsymbol{\beta}}$ is defined as:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\max}\, L(\boldsymbol{\beta}, \phi; \boldsymbol{y}) \tag{2.3}$$

Finding the values $\hat{\boldsymbol{\beta}}$ that maximize the likelihood corresponds to finding the values that maximize the log-likelihood $\ell(\boldsymbol{\beta}, \phi; \boldsymbol{y}) = \log(L(\boldsymbol{\beta}, \phi; \boldsymbol{y}))$. For the independence hypothesis on $Y_1, \ldots, Y_n$ we get:

$$\begin{aligned} \ell(\boldsymbol{\beta}, \phi; \boldsymbol{y}) &= \log(L(\boldsymbol{\beta}, \phi; \boldsymbol{y})) \\ &= \log\left(\prod_{i=1}^{n} \exp\left\{\frac{\omega_i}{\phi}\left[y_i\theta_i - b(\theta_i)\right]\right\} c(y_i, \phi, \omega_i)\right) \\ &= \sum_{i=1}^{n}\left\{\frac{\omega_i}{\phi}\left[y_i\theta_i - b(\theta_i)\right] + \log(c(y_i, \phi, \omega_i))\right\} \\ &= \sum_{i=1}^{n} \ell_i(\boldsymbol{\beta}, \phi; \boldsymbol{y}) \end{aligned} \tag{2.4}$$

The maximum value of $\ell(\boldsymbol{\beta}, \phi; \boldsymbol{y})$ can be obtained by imposing all its partial derivatives equal to 0:

$$\frac{\partial \ell(\boldsymbol{\beta}, \phi; \boldsymbol{y})}{\partial \beta_j} = 0, \quad \forall j \in \{0, 1, \ldots, p\}$$

These equations can be solved with numerical methods, such as Newton-Raphson algorithm or its variant Fisher scoring. It is possible to show that Newton-Raphson algorithm corresponds to iteratively solving a weighted least squares optimization problem.

In the case with Normal response and identity link, the optimization problem (2.3) has an explicit solution:

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^t \boldsymbol{X}\right)^{-1} \boldsymbol{X}^t \boldsymbol{y}$$

A statistic that can be used to measure the goodness of fit of a model is the *Deviance*. It can be used by comparing the current model log-likelihood $\ell\left(\hat{\boldsymbol{\beta}}, \phi; \boldsymbol{y}\right)$ with the *saturated model* log-likelihood $\ell_S\left(\boldsymbol{\beta}^*, \phi; \boldsymbol{y}\right)$. The saturated model is the model with n parameters, so a model where the expected values of the response variables $\mu_1, \ldots, \mu_n$ are estimated with their observed values $y_1, \ldots, y_n$. It is possible to show that $\ell_S\left(\boldsymbol{\beta}^*, \phi; \boldsymbol{y}\right) \geq \ell\left(\hat{\boldsymbol{\beta}}, \phi; \boldsymbol{y}\right)$. The closer $\ell\left(\hat{\boldsymbol{\beta}}, \phi; \boldsymbol{y}\right)$ is to $\ell_S\left(\boldsymbol{\beta}^*, \phi; \boldsymbol{y}\right)$, the better the current model fitting is.

Definition 2.2 (Deviance)**.** Given $\ell\left(\hat{\boldsymbol{\beta}}, \phi; \boldsymbol{y}\right)$ the log-likelihood of the current model and $\ell_S\left(\boldsymbol{\beta}^*, \phi; \boldsymbol{y}\right)$ the log-likelihood of the saturated model, the *Scaled Deviance* of the current model is defined as:

$$S(\hat{\boldsymbol{\beta}}, \phi, \boldsymbol{y}) = -2\left(\ell\left(\hat{\boldsymbol{\beta}}, \phi; \boldsymbol{y}\right) - \ell_S\left(\boldsymbol{\beta}^*, \phi; \boldsymbol{y}\right)\right)$$

The *Deviance* of the current model is defined as:

$$D(\hat{\boldsymbol{\beta}}, \boldsymbol{y}) = \phi\, S(\hat{\boldsymbol{\beta}}, \phi, \boldsymbol{y})$$

In deviance notation $D(\hat{\boldsymbol{\beta}}, \boldsymbol{y})$, the parameter ϕ is not reported because the deviance does not depend on ϕ. Indeed, from (2.4) we get:

$$\begin{aligned}
S(\hat{\boldsymbol{\beta}}, \phi, \boldsymbol{y}) &= -2\left(\ell\left(\hat{\boldsymbol{\beta}}, \phi; \boldsymbol{y}\right) - \ell_S\left(\boldsymbol{\beta}^*, \phi; \boldsymbol{y}\right)\right) \\
&= -2\left(\sum_{i=1}^{n}\left\{\frac{\omega_i}{\phi}\left[y_i\hat{\theta}_i - b(\hat{\theta}_i)\right] + \log\left(c(y_i, \phi, \omega_i)\right)\right\}\right. \\
&\qquad \left. - \sum_{i=1}^{n}\left\{\frac{\omega_i}{\phi}\left[y_i\theta_i^* - b(\theta_i^*)\right] + \log\left(c(y_i, \phi, \omega_i)\right)\right\}\right) \\
&= -2\left(\sum_{i=1}^{n}\frac{\omega_i}{\phi}\left\{\left[y_i\hat{\theta}_i - b(\hat{\theta}_i)\right] - \left[y_i\theta_i^* - b(\theta_i^*)\right]\right\}\right)
\end{aligned}$$

$$D(\hat{\boldsymbol{\beta}}, \boldsymbol{y}) = -2\left(\sum_{i=1}^{n}\omega_i\left\{\left[y_i\hat{\theta}_i - b(\hat{\theta}_i)\right] - \left[y_i\theta_i^* - b(\theta_i^*)\right]\right\}\right)$$

In table 2.3 the deviances for the families mentioned in 2.1 are reported.

As $\ell_S\left(\boldsymbol{\beta}^*, \phi; \boldsymbol{y}\right)$ does not depend on $\hat{\boldsymbol{\beta}}$, maximizing the likelihood in equation (2.3) is the same as minimizing the deviance, that can be seen as a *Loss Function*:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min}\, D(\boldsymbol{\beta}, \boldsymbol{y}) \tag{2.5}$$

Table 2.3: Deviance for Linear Exponential Families.

Distribution	**Deviance** $D(\hat{\boldsymbol{\beta}}, \boldsymbol{y})$
Normal	$\sum_{i=1}^{n} (y_i - \hat{\mu}_i)^2$
Poisson	$2 \sum_{i=1}^{n} \left\{ y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) - (y_i - \hat{\mu}_i) \right\}$
Gamma	$2 \sum_{i=1}^{n} \left\{ -\log\left(\frac{y_i}{\hat{\mu}_i}\right) + \frac{y_i - \hat{\mu}_i}{\hat{\mu}_i} \right\}$
Scaled Binomial	$2 \sum_{i=1}^{n} \left\{ y_i \log\left(\frac{y_i}{\hat{\mu}_i}\right) + (1 - y_i) \log\left(\frac{1-y_i}{1-\hat{\mu}_i}\right) \right\}$

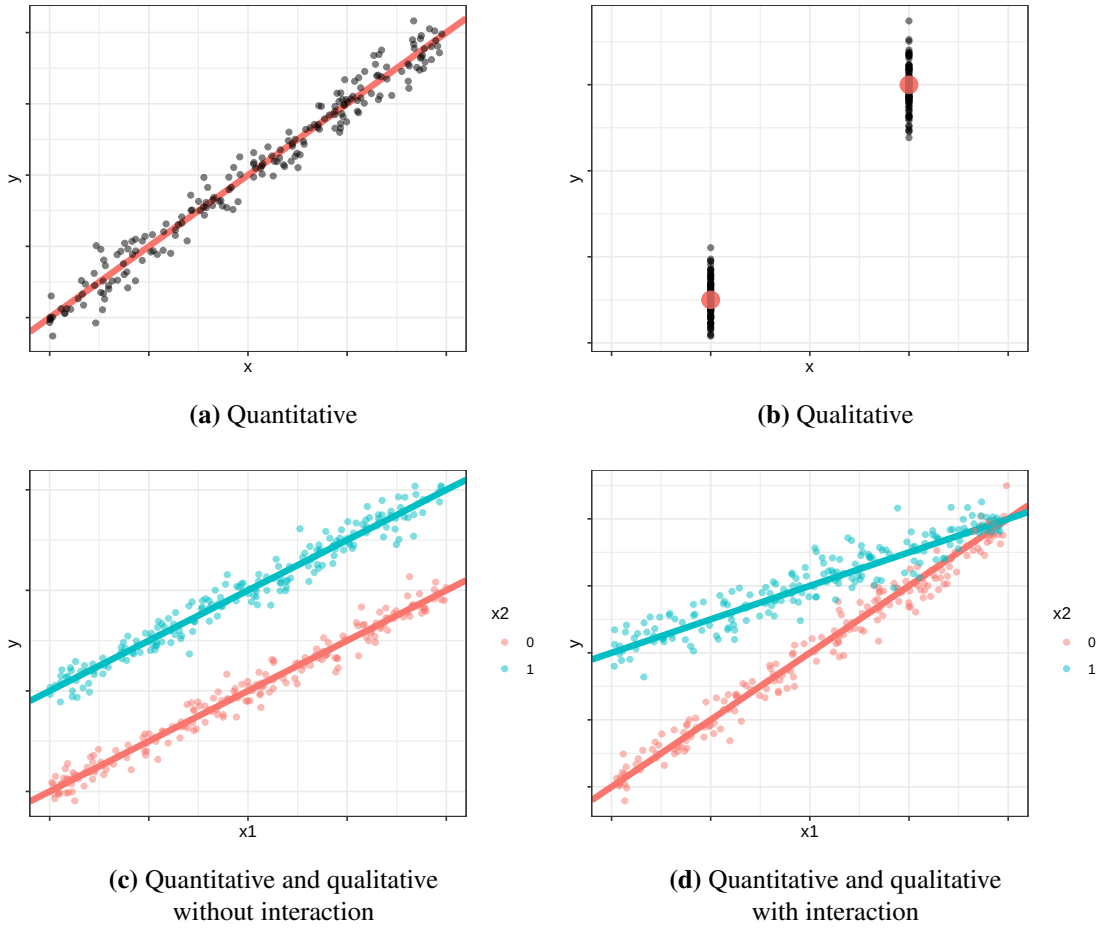
Variable effects

As we mentioned in 1.3.2, the explanatory variables can be *quantitative* or *qualitative*. In GLMs, if explanatory variables transformation terms aren't added to the linear predictor η, the variables effect on η is linear. In figure 2.2 the effects of quantitative and qualitative variables are shown. The data is simulated from a GLM with Normal response and identity link.

In the top-left panel, we see the effect of the quantitative variable x in the model $\mu_i = \beta_0 + \beta_1 x_i$. As we can see it is a straight line. The coefficient β_1 represents the slope of the line, thus $\beta_1 > 0$ means that x and Y are positively correlated, while $\beta_1 < 0$ means that x and Y are negatively correlated. For example, if x is the power of the vehicle and Y the yearly number of claims, $\beta_1 > 0$ means that the more powerful the vehicle is, the more claims the policyholder will experience on average.

In the top-right panel, we see the effect of a qualitative binary variable x in the model $\mu_i = \beta_0 + \beta_1 x_i$. The variable is encoded with values 0 and 1, so β_1 represents the effect of the modality $x = 1$. In general, for a qualitative variable with K modalities we will have $K - 1$ dummy variables $x'_1, \ldots, x'_{K-1}$ and the model will be $\mu_i = \beta_0 + \beta_1 x'_{i1} + \beta_2 x'_{i2} + \cdots + +\beta_{K-1} x'_{i,K-1}$. Thus, the β_j coefficient represents the relative effect of the modality j compared to the base level modality, that is the one not explicitly included in the dummy encoding. For example, if x is the vehicle make, Y the yearly number of claims, the base level for x is 'Fiat' and the j^{th} modality is 'Ferrari', then $\beta_j > 0$ means that Ferrari cars on average experience more claims that Fiat cars.

In general, in a multivariate model, the coefficient β_j represents the effect of the variable j given all the others. In the example of Fiat and Ferrari cars, if in the model there is also the variable 'vehicle power', the coefficient β_j corresponding to the modality 'Ferrari' represents the how more risky a Ferrari car is compared to a Fiat car with the same power. If the explanatory variables are strongly correlated, it is important to be



(a) Quantitative

(b) Qualitative

(c) Quantitative and qualitative without interaction

(d) Quantitative and qualitative with interaction

Figure 2.2: Explanatory variables types.

aware of this aspect. For example, Ferrari cars are usually more powerful that Fiat cars. So, it is possible that in general Ferrari cars are more risky than Fiat cars, but comparing a Ferrari car to a Fiat with the same power, the Ferrari could be less risky. This effect is called *Simpson's paradox* [1].

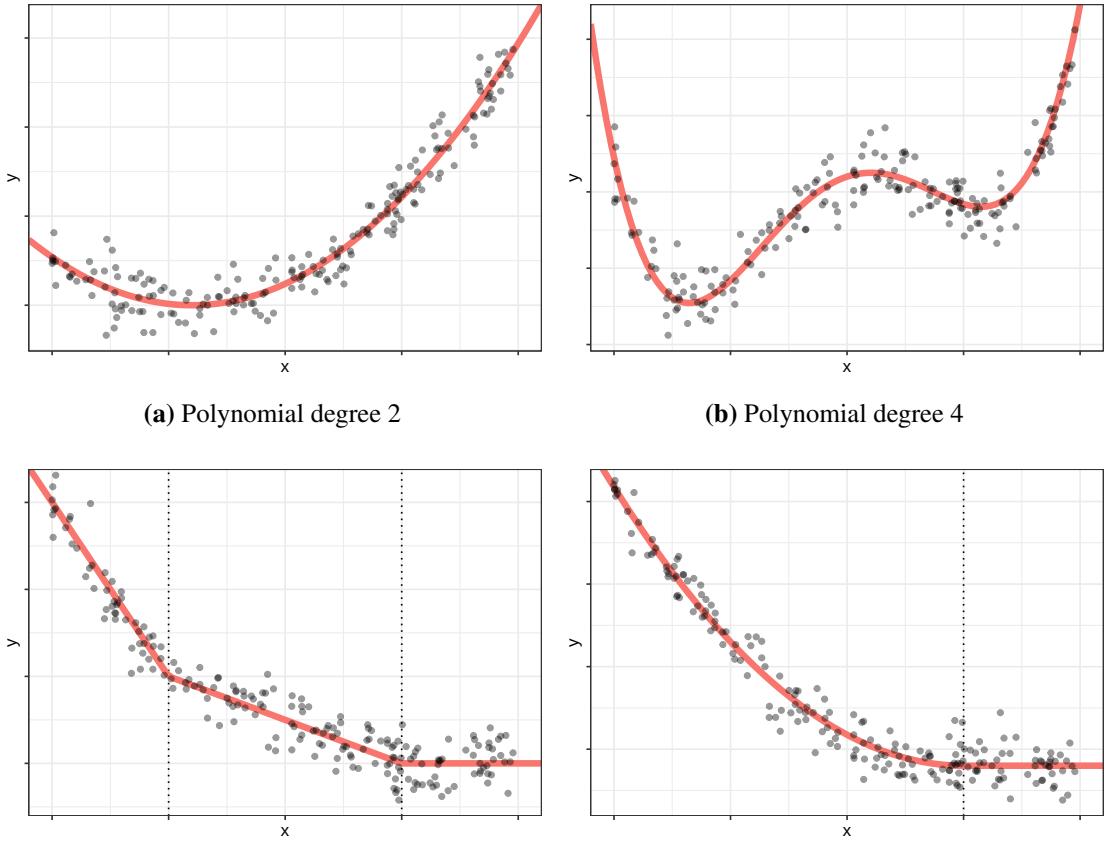
In the bottom-left panel of figure 2.2, we see the effect of a quantitative variable x_1 and a qualitative binary variable x_2 together in the model $\mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}$. As we can seen, the effects of x_1 variable in the two groups defined by x_2 variable are represented by two parallel straight lines. The first one is $\mu_i = \beta_0 + \beta_1 x_{i1}$ and the second is $\mu_i = (\beta_0 + \beta_2) + \beta_1 x_{i1}$. The coefficient β_2 represents the vertical distance between the two lines.

In the bottom-right panel, the interaction effect between x_1 and x_2 is included in the model. The model becomes $\mu_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i1} x_{i2}$. That means that the

[1] Simpson's paradox, `https://en.wikipedia.org/wiki/Simpson%27s_paradox`

effect of x_1 variables depends on the determination of the x_2 variable. In the group with $x_2 = 0$ the effect is represented by the line $\mu_i = \beta_0 + \beta_1 x_i$; the group with $x_2 = 1$ the effect is represented by the line $\mu_i = (\beta_0 + \beta_2) + (\beta_1 + \beta_3)\, x_{i1}$.

For quantitative variables, it is possible to consider also non linear effects in GLMs. Some examples are reported in figure 2.3.



(a) Polynomial degree 2

(b) Polynomial degree 4

(c) Piece-wise linear

(d) Piece-wise polynomial degree 2

Figure 2.3: Explanatory quantitative variables effects.

The basic way to achieve it is by adding polynomial terms to the linear predictor. For instance, if x is a quantitative variable, it is possible to add to the model the term x^2, obtaining the model $\mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2$. An example of model with both x and x^2 terms is represented in top-left panel of figure 2.3. Adding the quadratic term, the effect graph becomes a parabola.

With the same logic, it is possible to add more power terms. In general, if we want to model x with a polynomial of degree d, we can consider the model $\mu_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \cdots + \beta_d x_i^d$. In top-right panel of figure 2.3 a 4^{th} degree polynomial effect is represented. We highlight that the model is still considered linear, as the attribute

"Linear" in "General Linear Model" is referred to the relation between the parameters β_j and the linear predictor η_i that is still linear.

Another way to model non liner effects of explanatory variables is to separate the effects by pieces. In bottom-left panel of figure 2.3 a case in which the x effect is separated in 3 pieces is represented. As in all the pieces the effect is linear, the graph of the variable effect is a broken line. This effect can be achieved by adding to the model the terms $(x-\nu)_+$, where $(x)_+$ represents the positive part of x ($(x)_+ = \max(0,x)$) and ν is the value of x in the angular point. The ν values are called *knots*, If the knots are $\nu_1, \nu_2, \ldots, \nu_m$, the model can be represented as $\mu_i = \beta_0 + \beta_1 x_i + \beta_2(x_i - \nu_1)_+ + \beta_3(x_i - \nu_2)_+ + \cdots + \beta_{m+1}(x_i - \nu_m)_+$. This kind of functions are called *linear splines* and will be further discussed in section 2.1.2. If we want the effect to be null from a certain point ν, we can consider the variable $x' = \min(\nu, x)$ instead of x. This corresponds to aggregate to ν all the x after ν.

The piece-wise approach can be enhanced by also considering polynomial terms. For instance, in bottom-left panel of figure 2.3, the model represented is $\mu_i = \beta_0 + (x_i - \nu)_-^2$, where $(x)_-$ is the negative part of x ($(x)_- = \min(0,x)$). $f(x) = (x-\nu)^2$ is a parabola with vertex in ν. The fact of not adding the linear term leads to a monotonic effect made by a semi-parabola and a horizontal semi-line that starts from its vertex.

The examples represented in figures 2.2 and 2.3 are based on simulated data. That means that the linear predictor structure is known and the coefficients $\beta_0, \beta_1, \ldots, \beta_J$ are known. In practice, the real model is not known and the coefficients and the structure must be estimated by the data. Thus, we can take assumptions on the structure and we can estimate the coefficients with $\hat{\beta}_0, \hat{\beta}_1, \ldots, \hat{\beta}_J$. In many cases it is not so clear whether to consider or not a variable and how to consider it. For example, with the same data both bottom-left and bottom-right models could work fine. In section 2.1.1 we are going to discuss some variable selection techniques for GLM.

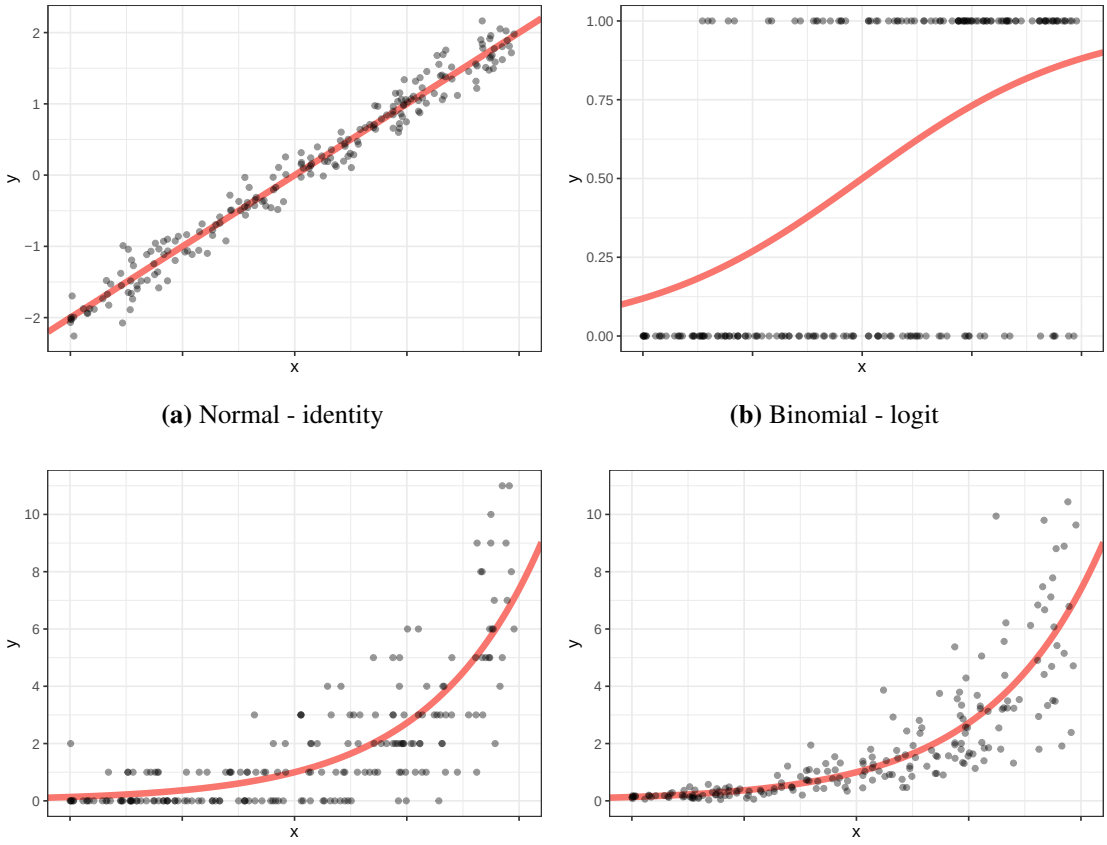
Link functions and relativities

As we mentioned in 2.1.1, GLM supports several families. In figure 2.4 the models $g(\mu_i) = \beta_0 + \beta_1 x_i$ with different response variable distributions and link functions are represented. As we can see from the plots, a linear effect on x corresponds to a logistic effect when the link is logit and to an exponential effect when the link is log.

If $g(\mu) = \log(\mu)$, the model structure can be expressed as:

$$\begin{aligned}\mu_i &= e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}} \\ &= e^{\beta_0}\left(e^{\beta_1}\right)^{x_{i1}}\left(e^{\beta_2}\right)^{x_{i2}}\ldots\left(e^{\beta_p}\right)^{x_{ip}}\end{aligned}$$

The term e^{β_j} can be seen as the multiplicative factor corresponding to the variable x_j. If x_j is a dummy variable, e^{β_j} is the factor the expected value μ_i is multiplied by when $x_{ij} = 1$. If x_j is a quantitative variable, e^{β_j} is the factor the expected value μ_i



(a) Normal - identity

(b) Binomial - logit

(c) Poisson - log

(d) Gamma - log

Figure 2.4: Response variables and link functions.

is multiplied by for every one-unit increasing of x_{ij}. Indeed:

$$\left(e^{\beta_j}\right)^{x_j+1} = e^{\beta_j}\left(e^{\beta_j}\right)^{x_j}$$

The fact that with a log link the relation between coefficients $\beta_0, \beta_1, \ldots, \beta_p$ and expected value μ_i becomes multiplicative is particularly useful to deal with exposure v_i. In section 1.3.4 we have seen that often the observations are couples (policy, accounting year), so they have different exposures v_i. Thus, we usually work with the number of claims occurred in the exposure period M_i and we observe its realization m_i. The assumption we take is that:

$$M_i \sim Poisson(v_i\mu_i)$$

where μ_i is the expected value of the yearly number of claims N_i.

Under the GLM assumptions, we obtain

$$\begin{aligned} E(M_i) &= v_i\mu_i = v_i e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}} \\ &= e^{\log(v_i)} e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}} \\ &= e^{\beta_0+\beta_1 x_{i1}+\cdots+\beta_p x_{ip}+\log(v_i)} \end{aligned}$$

That means that we can model $M_1, M_2, \ldots, M_n$ as response variables in a GLM with Poisson response in which the linear predictor depends on an offset additive term $\log(v_i)$.

If $g(p) = \text{logit}(p)$, the model structure can be expressed as:

$$\begin{aligned} \frac{p_i}{1-p_i} &= e^{\beta_0+\beta_1 x_{i1}+\beta_2 x_{i2}+\cdots+\beta_p x_{ip}} \\ &= e^{\beta_0} \left(e^{\beta_1}\right)^{x_{i1}} \left(e^{\beta_2}\right)^{x_{i2}} \ldots \left(e^{\beta_p}\right)^{x_{ip}} \end{aligned}$$

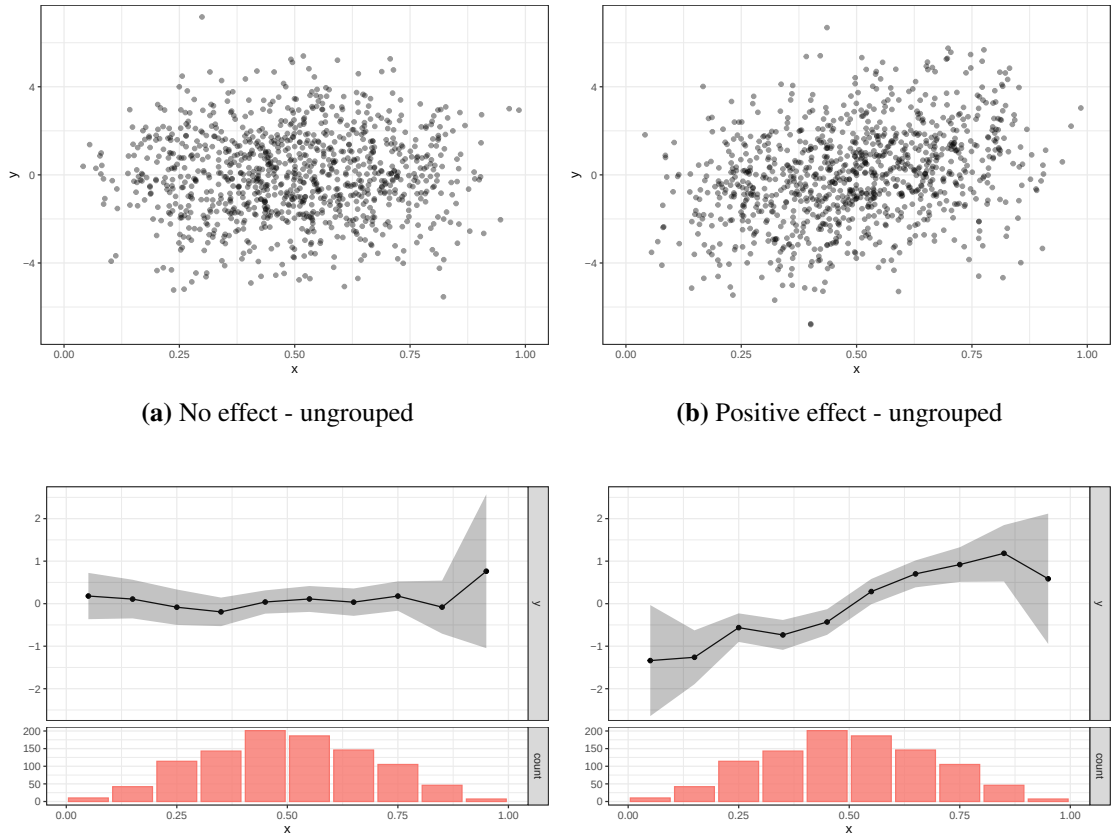
Thus, the term e^{β_j} can be seen as the multiplicative relativity corresponding to the variable x_j. However, in this case the relativity doesn't multiply directly the probability of success p, but it multiplies the odds of success $\frac{p}{1-p}$.

Variable selection

One of the most challenging aspects of GLM fitting is selecting the variables and their effects by looking to observed data. In practice, we usually have many explanatory variables available but only some of them are relevant for the prediction of the response variable. Adding useless variables to the model increases the variance of the estimators of the coefficients $\tilde{\beta}_0, \tilde{\beta}_1, \ldots, \tilde{\beta}_p$ and then the variance of the predictions $\tilde{\mu}_i = \tilde{\beta}_0 + \tilde{\beta}_1 x_{i1} + \cdots + \tilde{\beta}_p x_{ip}$. On the other hand, being too frugal with explanatory variables could lead to wasting part of the predictive power of the available explanatory variables.

One useful tool we have to understand if an explanatory variable x is relevant or not is plotting the points (x_i, y_i), as we did in 2.2, 2.3 and 2.4. If there are too many observations and the plot is not easily readable, it is possible to group the points by x modalities and show for each group the average of y_i and a confidence interval that gives an idea on the dispersion of the observations around the average. If x is a continuous variable with too many modalities, it is possible to group them into buckets. Showing the average of y_i for groups of x is particularly useful for Binomial and Poisson data, where the fact that y_i can present few different values compromises the plot readability. An example is reported in figure 2.5. The top-left and bottom-left panels represent a case in which x and y are not related, while the top-right and bottom-right panels represent a case of positive correlation. From the ungrouped plot in the top-right panel the effect is not clear, while from bottom-right panel it is evident.

If we are dealing with a multivariate model where we already inserted the variables $x_1, \ldots, x_p$ and we want to evaluate the additional information brought by x_{p+1}, it is



(a) No effect - ungrouped

(b) Positive effect - ungrouped

(c) No effect - grouped

(d) Positive effect - grouped

Figure 2.5: Explanatory variable effect evaluation. The top-left and bottom-left panels represent a case in which x and y are not related, while the top-right and bottom-right panels represent a case of positive correlation. From the ungrouped plot in the top-right panel the effect is not clear, while from bottom-right panel it is evident.

possible to look at the plot $(x_{i,p+1}, r_i)$, where r_i are the residuals of the model without the variable x_{p+1}:

$$r_i = y_i - \hat{\mu}_i = y_i - g^{-1}\left(\hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \cdots + \hat{\beta}_p x_{ip}\right), \quad i \in \{1, 2, \ldots, n\}$$

If the plot shows a clear trend, we will add the variable x to the model, otherwise we won't.

The choice of adding or not a variable in the model can be supported by *hypothesis testing*. Given a GLM with coefficients $\beta_0, \beta_1, \ldots, \beta_p$, it is possible to test if a group of coefficients $\beta_{j_1}, \beta_{j_2}, \ldots, \beta_{j_s}$ is equal to zero. Formally the hypotheses tested are:

$$\begin{cases} H_0: & \beta_{j_k} = 0 \; \forall k \in \{1, 2, \ldots, s\} \\ H_1: & \exists k : \beta_{j_k} \neq 0 \end{cases}$$

If the hypothesis H_0 is accepted, the variables $x_{j_1}, x_{j_2}, \ldots, x_{j_s}$ can be removed from the model; if the hypothesis H_0 is rejected, at least some of the variables $x_{j_1}, x_{j_2}, \ldots, x_{j_s}$ should be kept in the model.

If we want to test if a quantitative variable x_j has a significant effect, we can conduct the test on the single coefficient β_j. If we want to test if a qualitative variable with dummy encoding $x_{j_1}, x_{j_2}, \ldots, x_{j_s}$ is significant, we can conduct the test on the coefficients $\beta_{j_1}, \beta_{j_2}, \ldots, \beta_{j_s}$. For qualitative variables it is also possible to conduct a test for each level x_{j_k}; in this case we would test one by one if each level has an effect that is significantly different form the base level effect.

To conduct the test it is possible to use several statistics. One of them is the test based on *log likelihood ratio*. Let's indicate with $\hat{\boldsymbol{\beta}}$ the estimated coefficients from the model without any constraint and with $\hat{\boldsymbol{\beta}}^{(0)}$ the estimated coefficients with the constraints defined by H_0. As the space $\hat{\boldsymbol{\beta}}^{(0)}$ belongs to is a subset of the space $\hat{\boldsymbol{\beta}}$ belongs to, it results:

$$L\left(\hat{\boldsymbol{\beta}}^{(0)}\right) \leq L\left(\hat{\boldsymbol{\beta}}\right)$$

and then:

$$\frac{L\left(\hat{\boldsymbol{\beta}}^{(0)}\right)}{L\left(\hat{\boldsymbol{\beta}}\right)} \leq 1$$

The basic idea is that, if the variables $x_{j_1}, x_{j_2}, \ldots, x_{j_s}$ have a significant effect, $L\left(\hat{\boldsymbol{\beta}}\right)$ will be much higher than $L\left(\hat{\boldsymbol{\beta}}^{(0)}\right)$ and we will reject the hypothesis H_0, while if the variables $x_{j_1}, x_{j_2}, \ldots, x_{j_s}$ have not a significant effect, $L\left(\hat{\boldsymbol{\beta}}\right)$ will be more or less the same as $L\left(\hat{\boldsymbol{\beta}}^{(0)}\right)$ and we will accept the hypothesis H_0.

To perform the test, the quantity usually employed is the following:

$$\begin{aligned}\lambda &= -2\log\left(\frac{L\left(\hat{\boldsymbol{\beta}}^{(0)}\right)}{L\left(\hat{\boldsymbol{\beta}}\right)}\right) \\ &= -2\left[\ell\left(\hat{\boldsymbol{\beta}}^{(0)}\right) - \ell\left(\hat{\boldsymbol{\beta}}\right)\right]\end{aligned}$$

If we indicate $\tilde{\lambda} = -2\left[\ell\left(\tilde{\boldsymbol{\beta}}^{(0)}\right) - \ell\left(\tilde{\boldsymbol{\beta}}\right)\right]$, it is possible to demonstrate that, under the hypothesis H_0, $\tilde{\lambda}$ has approximately chi-squared distribution with s degrees of freedom:

$$\tilde{\lambda} \dot{\sim} \chi^2(s)$$

Therefore, with a significance level α we will reject H_0 when $\lambda > \chi_{s,1-\alpha}$, where $\chi_{s,1-\alpha}$ is the quantile of order $1-\alpha$ of the distribution $\chi^2(s)$.

The same approach can be used in general for testing hypotheses that can be expressed as $H_0 : \boldsymbol{L}\boldsymbol{\beta} = \xi$, where $\boldsymbol{L}$ is a matrix $s \times (p+1)$ and $\xi \in \mathbb{R}^{p+1}$. This is particularly useful in qualitative variables to test if some of the levels have the same effect and can be then unified. For example, if x_{j_1} and x_{j_2} are two dummy variables that describe two levels of the same quantitative variable, we can perform the test $H_0 : \beta_{j_1} = \beta_{j_2}$ in order to decide whether unifying the two levels is suitable or not.

Anyway, selecting the variables by performing hypotheses testing have some drawbacks. First of all, conducting a lot of test produce the multiple test problem. Let's consider the case in which we conduct a test of the kind $H_0 : \beta_j = 0$ with a significance level $\alpha = 0.05$ on many variables that have no effect on the response. On average, although the null hypotheses are always true, we will reject them once every 20 tests. That means that if we have available 100 variables, we will randomly select 5 of them, falling in *overfitting*. To fix this problem, it is possible to use the Bonferroni correction, that consist in dividing α by the number of test conducted to define the rejection region. But this could be a too restrictive correction that could lead to discard from the model some useful variables, falling in *underfitting*.

Moreover, hypothesis testing aim is finding whether data supports or not an hypothesis. If the aim of the model is prediction, basing variable selection on hypotheses testing could lead to a sub-optimal model.

Another method for variable selection is comparing the models by computing *information criteria*. Two information criteria commonly used are the *Akaike Information Criterion* (AIC) and the *Bayesian Information Criterion* (BIC):

$$AIC = -2\ell(\boldsymbol{\beta}) + 2(p+1)$$
$$BIC = -2\ell(\boldsymbol{\beta}) + \log(n)(p+1)$$

The aim of these statistics is to penalize the likelihood by adding a component that measures the complexity of the model. Among the models considered, the optimal model will be the one that minimizes the information criterion. Thus, if two models have the same likelihood, the optimal model will be the one with less parameters. If the model with more parameter has a higher likelihood, it will be chosen only if that increase in likelihood compensated the increase in complexity.

Another way to compute model predictive performance is by randomly splitting the available dataset $\mathcal{D} = \{(\boldsymbol{x}_1, \omega_1, y_1), \ldots, (\boldsymbol{x}_n, \omega_n, y_n)\}$ into a *training set* (or learning set) $\mathcal{D}^{\mathcal{B}}$ and a *test set* $\mathcal{D}^{\bar{\mathcal{B}}}$, with $\mathcal{B} \subset \{1, 2, \ldots, n\}$ labeling the dataset $\mathcal{D}^{\mathcal{B}} = \{(\boldsymbol{x}_i, \omega_i, y_i) : i \in \mathcal{B}\} \subset \mathcal{D}$ and $\bar{\mathcal{B}} = \{1, 2, \ldots, n\} \setminus \mathcal{B}$. With this split, we can fit the model on only the training set $\mathcal{D}^{\mathcal{B}}$ and assess its performance on the test set $\mathcal{D}^{\bar{\mathcal{B}}}$ by computing the deviance $D\left(\hat{\boldsymbol{\beta}}^{\mathcal{B}}, \boldsymbol{y}^{\bar{\mathcal{B}}}\right)$, where $\hat{\boldsymbol{\beta}}^{\mathcal{B}}$ is the vector of the coefficients estimated on the training set and $\boldsymbol{y}^{\bar{\mathcal{B}}}$ is the vector of the observed response variables in

the test set. This way it is possible to choose the best model as the one that minimizes the deviance in the test set and then fitting it with the whole dataset $\mathcal{D}$.

A limit of the train-test approach is that it could bring to overfitting in the test set. Indeed, in particular if the dataset is small, it is possible that a specific set of variables minimizes the deviance on the test set just by chance. To prevent this, it is possible to conduct a *K-fold cross validation*. This consists in randomly partitioning the dataset $\mathcal{D}$ into K subsets $\mathcal{D}^{\mathcal{B}_1}, \mathcal{D}^{\mathcal{B}_2}, \dots, \mathcal{D}^{\mathcal{B}_K}$ and, for each subset $\mathcal{D}^{\mathcal{B}_k}$, performing a train-test procedure keeping $\mathcal{D}^{\setminus\mathcal{B}_k} = \mathcal{D} \setminus \mathcal{D}^{\mathcal{B}_k}$ as a training set and $\mathcal{D}^{\mathcal{B}_k}$ as a test set. For each k we can compute the testing deviance $D\left(\hat{\boldsymbol{\beta}}^{\setminus\mathcal{B}_k}, \boldsymbol{y}^{\mathcal{B}_k}\right)$. We can then compute the average deviance within the K subset as:

$$D^{CV(K)} = \frac{1}{K}\sum_{k=1}^{K} D\left(\hat{\boldsymbol{\beta}}^{\setminus\mathcal{B}_k}, \boldsymbol{y}^{\mathcal{B}_k}\right)$$

Thus, the best model will be the one that minimize $D^{CV(K)}$.

The higher K is, the less subjected to randomness $D^{CV(K)}$ is. However, the higher K is, the more computationally expensive the procedure is. A common choice for K is $K = 10$. If we choose $K = n$, the procedure is also called *leave-one-out cross validation*.

Scalability and manual fitting

One problem of GLM is that the variables selection process is not so easily scalable. Indeed, if we consider p explanatory variables, even without taking into account interactions and quantitative variables transformation, there are 2^p possible models that can be obtained by choosing a subset of those variables. As p increases, building all these models and choosing the optimal one becomes unfeasible.

One strategy to reduce the time consumption is to use a *stepwise procedure*. First of all we must choose a criterion to compare the models, such as the AIC. Then we have to choose a starting model that consider the variables subset $\mathcal{C}_0 \subset \{1, 2, \dots, p\}$. It is possible to compare this model with all the models that can be obtained by removing one variable from $\mathcal{C}_0$ or adding to $\mathcal{C}_0$ one that is not included in it. From all these models we can compute the AIC and we will choose as $\mathcal{C}_1$ the set of variables that minimize the AIC. If none of the considered models has an AIC lower to the one obtained with the variables $\mathcal{C}_0$, the procedure ends and our final set of variables is $\mathcal{C}_0$. Otherwise, we will repeat the step with $\mathcal{C}_1$. The procedure can be iteratively repeated until we obtain a subset of variables $\mathcal{C}_f$ that can't be improved by removing or adding a variable. The model with the variables $\mathcal{C}_f$ will be our chosen one.

This procedure is much faster than computing all the 2^p models, but it is still not so scalable for large p. Moreover, in this procedure we are not taking into account the interactions and the possible transformations for the quantitative variables. This can be achieved by slightly modifying the algorithm, but it would further increase the

complexity and the computation time. Another option is starting from the result of the stepwise regression and manually choosing interactions and quantitative variables transformations by looking at plots as described in 2.1.1.

One characteristic of GLM is that the variables effects can be easily interpreted and the variable selection process is for large part manual. This aspect can be problematic if there are a lot of explanatory variables, but it brings some important benefits too. Indeed, the actuary can take choices on variable selection not only based on observed data, but also on his domain knowledge. For instance, the choice of selecting or not an explanatory variable can be guided also on its interpretation: if the observed effect makes sense, it can be added to the model even if it is not statistically significant and it doesn't decrease the AIC, and, on the opposite, if the observed effect is not reasonable, the actuary can choose not to include the variable in the model even if its effect is supported by the data. So, in the actuarial practice, in the GLM fitting process there is always a subjective component that impacts on the final result. For these reason it is important for the actuary to have a deep knowledge on the phenomenon he is modeling.

An aspect that facilitates the model building and reduces the complexity of the process, even if there are many variables, is that usually the models are not built from scratch. Actuarial models are usually updated once a year, so it is also possible to start the new model by fitting on new data the final model from the year before. As the models are usually built with policies data from more than one year, the new dataset partially overlaps with the one from the previous year. However, the overlapping observations aren't identical: the new dataset will have the new settlement information and some new explanatory variables. Anyway, if the actuary is familiar with the effects of the variables in the previous models, he already knows which will probably be relevant and which don't even deserve too much attention.

2.1.2 Generalized Additive Models

In section 2.1.1 we have seen that sometimes quantitative variables have not a linear effect on the linear predictor. In GLM it is possible to deal with non-linear effects by adding polynomial or split-wise terms. GAMs are models based on GLM that introduce a more flexible way to deal with quantitative variables with non-linear effects.

Model assumptions

In GAM, the assumptions are the same of GLM, stated in 2.1.1, with the following advancement in the linear predictor:

$$g(\mu_i) = \eta_i = \boldsymbol{x}_i^t \boldsymbol{\beta} + \sum_{l=1}^{q} f_l(z_{i,l}), \qquad i \in \{1, 2, \dots, n\}$$

where

- $\boldsymbol{x}_i$ is the vector of the variables with a linear effect as described in GLM, that also include a term 1 that represents the intercept;
- $\boldsymbol{\beta}$ is the vector of the linear coefficients as described in GLM;
- $z_{i,1}, z_{i,2}, \dots, z_{i,q}$ are the quantitative variables with a non linear effect;
- $f_1(\cdot), f_2(\cdot), \dots, f_q(\cdot)$ are continuous functions with continuous first and second derivatives.

The functions $f_l(\cdot)$ introduce the possibility to model non-linear effects of the variables z_l.

Cubic splines

A class of functions commonly used for modeling $f_l(\cdot)$ is the class of the *cubic splines*.

Definition 2.3 (Cubic Splines)**.** Let consider m real numbers $\nu_1, \nu_2, \dots, \nu_m$, called *knots*, and a function $f : [\nu_1, \nu_m] \to \mathbb{R}$ such that:

$$f(x) = h_t(x), \qquad x \in [\nu_t, \nu_{t+1}[$$

where, for $t = 1, 2, \dots, m-1$, $h_t(x) = \alpha_t + \vartheta_t x + \gamma_t x^2 + \delta_t x^3$. For the last index $m-1$, $f(x) = h_{m-1}(x)$ is extended to $[\nu_{m-1}, \nu_m]$.

$f(x)$ is a *cubic spline* if it satisfies the following conditions in the internal knots $\nu_2, \nu_3, \dots, \nu_{m-1}$

$$h_{t-1}(\nu_t) = h_t(\nu_t), \qquad h'_{t-1}(\nu_t) = h'_t(\nu_t), \qquad h''_{t-1}(\nu_t) = h''_t(\nu_t) \tag{2.6}$$

The constraints for $f(\cdot)$ make it a continuous functions with first and second derivatives continuous in $]\nu_1, \nu_m[$. It is possible to extend the cubic spline $f(\cdot)$ to

an interval $[a, b] \supset [\nu_1, \nu_m]$ with linear extensions on $[a, \nu_1[$ and $]\nu_m, b]$. The so built function $f : [a, b] \to \mathbb{R}$ is called *natural cubic spline*.

In definition 2.3 we introduced 4 parameters $(\alpha_t, \vartheta_t, \gamma_t, \delta_t)$ for each of the $m-1$ intervals $[\nu_t, \nu_{t+1}[$, $t \in \{1, 2, \ldots, m-1\}$. So, we have $4(m-1)$ parameters. In equation (2.6) we introduced 3 constraints for each of the $m-2$ knot in $\nu_2, \nu_3, \ldots, \nu_{m-1}$. So the free parameters become $4(m-1) - 3(m-2) = m+2$. The linear extension on $[a, \nu_1[$ and $]\nu_m, b]$ corresponds to the constraint $f''(x) = 0$ on $[a, \nu_1[\cup]\nu_m, b]$, thus, $h_1''(\nu_1) = 0$ and $h_m''(\nu_m) = 0$. Adding these two constraints, we get that the natural cubic spline $f : [a, b] \to \mathbb{R}$ has m degrees of freedom.

With an approach similar to the one we used in 2.1.1 for split-wise effects in GLM, we can represent a cubic spline by the functions $x \mapsto (x - \nu_t)_+^3$, $t = 1, 2, \ldots, m$.

The expression:

$$f(x) = \alpha_0 + \vartheta_0 x + \sum_{t=1}^{m} c_t \, (x - \nu_t)_+^3, \qquad \text{with} \quad \sum_{t=1}^{m} c_t = 0 \quad \text{and} \quad \sum_{t=1}^{m} c_t \nu_t = 0 \tag{2.7}$$

gives a natural cubic spline.

The two side constraints ensure that we have a smooth linear extension right of ν_m. The expression (2.7) presents $m+2$ parameters, thus, with the 2 side constraints, there are m degrees of freedom. From this expression it is easy to show that the natural cubic splines over the interval $[a, b]$ with knots $\nu_1, \nu_2, \ldots, \nu_m$ constitute a m-dimensional vectorial space.

Smoothing

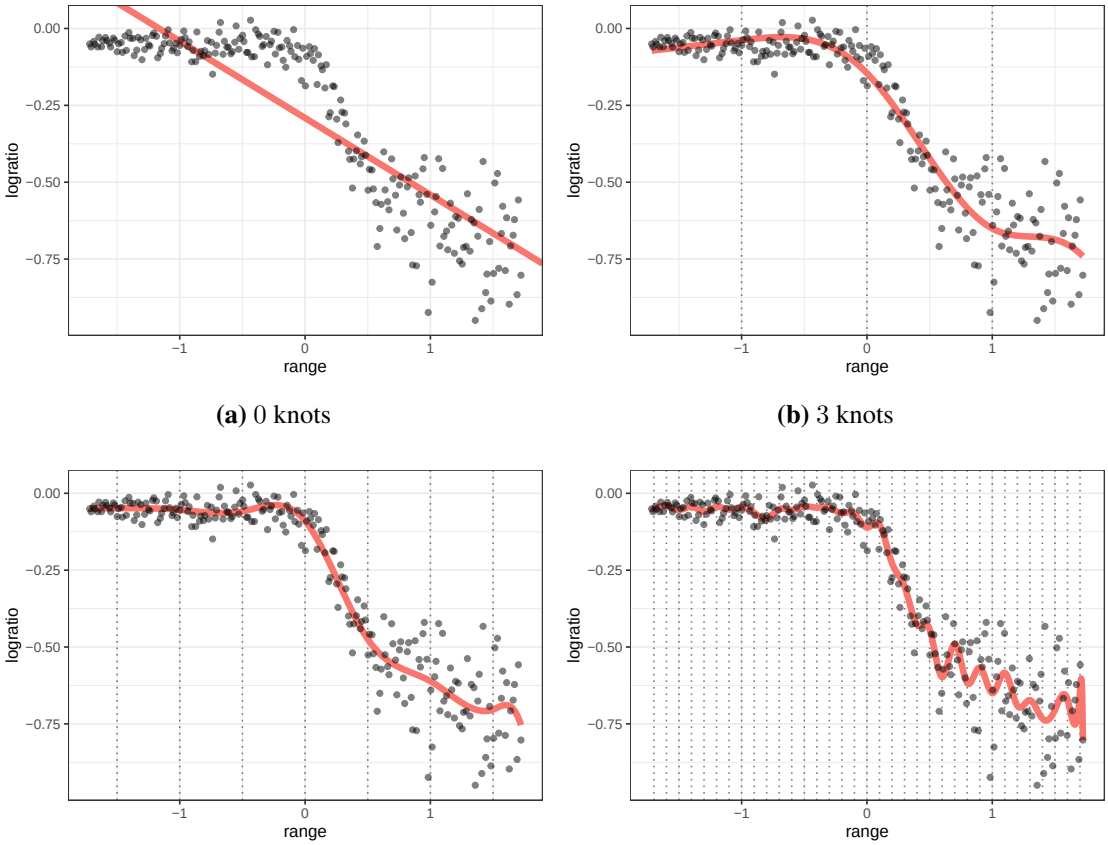
If we try to fit a cubic spline on data, we find that by increasing the number of knots, the function tends to overfit data. This is due to the fact that, increasing the number of knots, we are increasing the number of parameters of the model and, thus, the variance of the parameters estimators increases. We can see this effect in figure 2.6.

In the top-left panel, the model fitted is just linear and it clearly doesn't follow the path made by the observed points. In the bottom-right panel, the models fitted is a spline with 35 knots and it is clearly too wiggly compared to the path made by the observed points.

A measure of the wiggliness of a function is given by the integral of its squared second derivative:

$$\int_a^b \left(f''(x)\right)^2 dx \tag{2.8}$$

In figure 2.7 some examples of functions $f(x)$ and their squared second derivatives $\left(f''(x)\right)^2$ are represented. In the top panels, four functions of increasing wiggliness are represented and, in the bottom panels, we can see their four squared second derivatives.



(a) 0 knots **(b)** 3 knots

(c) 7 knots **(d)** 35 knots

Figure 2.6: GLM with cubic splines for different numbers of knots. The more knots there are, the wigglier the estimated function $\hat{f}(x)$ is.

As we can see from the plots, if a function $f(x)$ is linear, $(f''(x))^2$ is null and, as the curvature of the function increases, $(f''(x))^2$ increases. For this reason, $(f''(x))^2$ can also be seen as a measure of how much $f(x)$ differs from a linear function.

What is done in GAM to contain this wiggliness and to prevent overfitting is taking into account a penalization based on $(f''(x))^2$. If we have a regression problem with one explanatory variable x, this can be achieved by considering the minimization problem in (2.5) and adding a penalization term to the deviance:

$$\hat{f} = \arg\min_f \left\{ D(f, \boldsymbol{y}) + \lambda \int_a^b (f''(x))^2 \, dx \right\} \tag{2.9}$$

In the notation of the formula (2.9) we used f to indicate all the parameters of the natural cubic spline. The hyper-parameter $\lambda \geq 0$ is called *smoothing parameter*. It measures how much we want to penalize wiggliness. If we choose $\lambda = 0$ we won't penalize for $(f''(x))^2$ and the optimization problem corresponds to the maximum

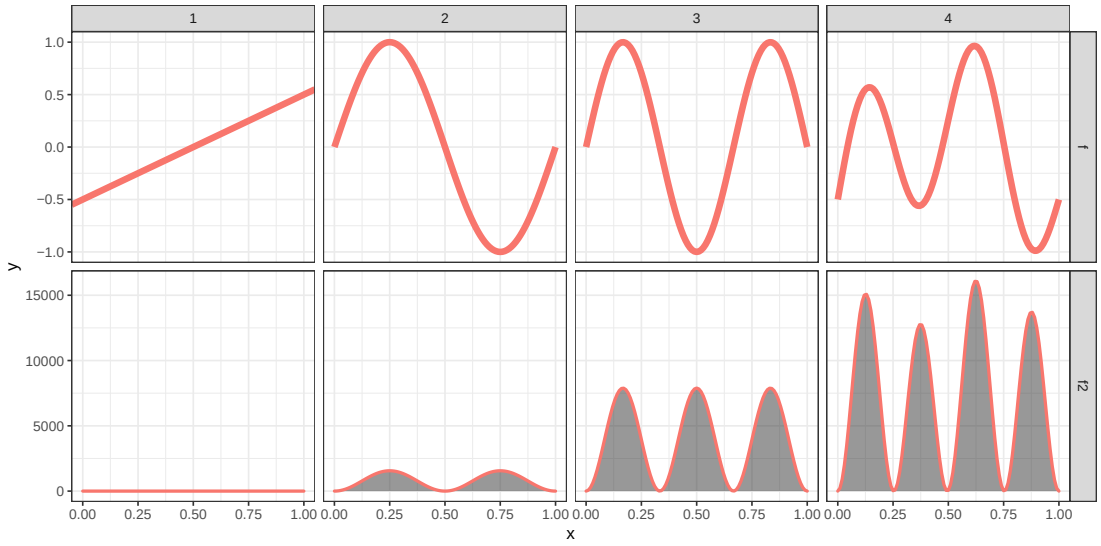


Figure 2.7: Squared second derivative $(f''(x))^2$ for functions with different wiggliness. The wigglier the function $f(x)$ is, the higher $(f''(x))^2$ is.

likelihood. The higher λ is, the more penalization for $(f''(x))^2$ we introduce and the smoother the estimate $\hat{f}(x)$ will be. If $\lambda \to +\infty$, we will introduce an infinite penalization for wiggliness, so the result will have $\hat{f}''(x) = 0$, thus $\hat{f}(x)$ will be linear. An example with different levels of λ can be seen in figure 2.8.

All the models have been fitted with $m = 50$ knots. As we can see, with $\lambda = 0$ we are clearly overfitting observations, while with $\lambda = 10^6$ we are clearly underfitting them.

As soon as the number of knots m is large enough, the exact number m and the positioning of the knots $\nu_1, \nu_2, \ldots, \nu_m$ is not important. If the function is flexible enough, the tuning of the curve is done by just tuning λ. It is possible to use as many knots as the determinations of x are, but it could be too computationally expensive. A possible choice is to fix a number of knots m and positioning $\nu_1, \nu_2, \ldots, \nu_m$ on empirical quantiles of the observed x or equally spaced in the range of x. In our example, $m = 50$ knots seems to be large enough.

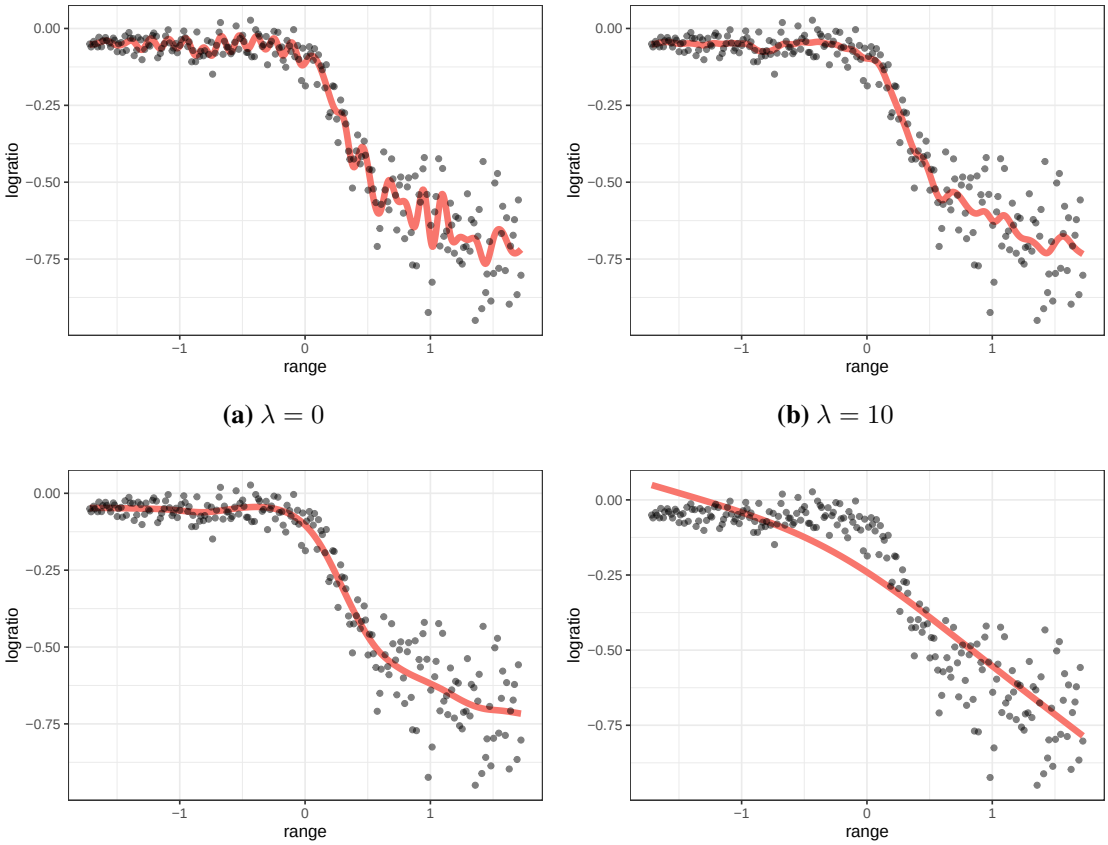
In general, if we have q quantitative explanatory variables that we want to fit with splines, the formula (2.9) becomes:

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \left\{ D(\boldsymbol{f}, \boldsymbol{y}) + \sum_{l=1}^{q} \lambda_l \int_{a_l}^{b_l} \left(f_l''(x_l)\right)^2 dx \right\} \tag{2.10}$$

where $\boldsymbol{f} = (f_1, f_2, \ldots, f_q)$.

In this context, we have to tune a vector of q smoothing parameters:

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \ldots, \lambda_q)$$



(a) $\lambda = 0$ **(b)** $\lambda = 10$

(c) $\lambda = 10^3$ **(d)** $\lambda = 10^6$

Figure 2.8: GAM estimate $\hat{f}(x)$ for different levels of the smoothing parameter λ. The higher λ is, the less wiggly the estimated function $\hat{f}(x)$ is.

Choice of the smoothing parameters

As described, the selection of the x_l effect consists just in finding the optimal parameter λ_l. The technique commonly used in machine learning for hyper-parameter tuning is the *Cross Validation*.

For a set Λ of values of $\boldsymbol{\lambda}$ we can perform a K-fold cross validation as described in 2.1.1 and, for each $\boldsymbol{\lambda} \in \Lambda$, we can compute the average test deviance:

$$D_{\boldsymbol{\lambda}}^{CV(K)} = \frac{1}{K} \sum_{k=1}^{K} D\left(\hat{\boldsymbol{f}}_{\boldsymbol{\lambda}}^{\setminus \mathcal{B}_k}, \boldsymbol{y}^{\mathcal{B}_k}\right)$$

At this point, we will choose the hyper-parameter vector that minimizes the cross-validation deviance:

$$\hat{\boldsymbol{\lambda}} = \arg\min_{\boldsymbol{\lambda} \in \Lambda} D_{\boldsymbol{\lambda}}^{CV(K)}$$

In particular if q is big, this procedure can be too computationally expensive. Two alternatives are the *Generalized Cross Validation* (GCV) and the *Un-Biased Risk Estimator* (UBRE). The idea behind these approaches is to estimate the test set deviance by just computing the training set deviance and applying to it a correction that penalizes for the complexity of the model, as it is done in AIC and BIC, described in section 2.1.1.

GCV and UBRE formulas are based on the fact that GAMs with identity link and Normal response are *linear smoothers*, i.e. $\hat{\boldsymbol{\mu}}$ can be expressed as:

$$\hat{\boldsymbol{\mu}} = H\boldsymbol{y} \tag{2.11}$$

where H is a $n \times n$ matrix that depends on the design matrix $\boldsymbol{X}$ and the hyper-parameters. H is called *smoothing matrix*. In general, if the link is not the identity and the response is not Normal, GAM is not a linear smoother, but the formula (2.11) still holds up with a local approximation.

It can be proven that the trace of H measures the flexibility of the function. Indeed, in Linear Model $H = \boldsymbol{X}\left(\boldsymbol{X}^t\boldsymbol{X}\right)^{-1}\boldsymbol{X}^t$ and $\text{tr}(H) = \text{tr}\left(\boldsymbol{X}\left(\boldsymbol{X}^t\boldsymbol{X}\right)^{-1}\boldsymbol{X}^t\right) = p + 1$, that is the number of *degrees of freedom* of the model. For this reason $\text{tr}(H)$ is also called number of *effective degrees of freedom* of the model. In GAM, as the smoothing parameter λ increases, the flexibility of the model decreases and $\text{tr}(H)$ decreases.

For linear smoothers it can be proved that the *leave one out cross validation* is:

$$D^{CV(n)} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{\mu}_i}{1 - H_{ii}}\right)^2$$

where H_{ii} is the i^{th} element of the diagonal of the smoothing matrix H.

This formula is particularly convenient because it requires just the fit with the whole dataset and not one fit for each fold. Indeed, in the formula the quantities that must be computed are μ_i and H_{ii}, that both depend on the model fitted with the whole dataset.

This formula can be further simplified by replacing the values H_{ii} with their average $\frac{\text{tr}(H)}{n}$. This way we get the GCV.

$$GCV = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \hat{\mu}_i}{1 - \frac{\text{tr}(H)}{n}}\right)^2 \tag{2.12}$$

As for K-fold cross validation we can compute the GCV for different values of $\boldsymbol{\lambda} \in \Lambda$ and choose the value $\hat{\boldsymbol{\lambda}}$ that minimizes the GCV.

The formula (2.12) can be expressed as:

$$\begin{aligned} GCV &= \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i-\hat{\mu}_i}{1-\frac{\text{tr}(H)}{n}}\right)^2 \\ &= \left(\frac{n}{n-\text{tr}(H)}\right)^2 \underbrace{\frac{\sum_{i=1}^{n}\left(y_i-\hat{\mu}_i\right)^2}{n}}_{=D(\hat{\boldsymbol{f}},\boldsymbol{y})} \\ &= \left(\frac{n}{n-\text{tr}(H)}\right)^2 D(\hat{\boldsymbol{f}},\boldsymbol{y}) \end{aligned} \tag{2.13}$$

The expression (2.13) generalizes the GCV to all the GAMs, also in the case in which the link is not identity and the response is not Normal.

To select the best value of $\boldsymbol{\lambda}$, we can perform a similar procedure using the UBRE in place of the GCV. The UBRE is defined as:

$$UBRE = \frac{1}{n}D(\hat{\boldsymbol{f}},\boldsymbol{y}) - \phi + \frac{2}{n}\text{tr}(H)\phi$$

Using the UBRE instead the GCV is preferred when ϕ is known, such as in Poisson regression case, in which $\phi = 1$.

Why cubic splines

In section 2.1.2 we said that *cubic splines* are commonly used for modeling $f(\cdot)$ in GAMs. This choice comes from the following theorem.

Theorem 2.1 (Spline property)**.** *Given the knots $\nu_1, \nu_2, \ldots, \nu_m$ and the values $y_1, y_1, \ldots, y_m$, for any $a \leq \nu_1$ and $b \geq \nu_m$, only one natural cubic spline $s(\cdot)$, such that $s(\nu_k) = y_k,\ k \in \{1, \ldots, m\}$, exists.*

Moreover, given $f(\cdot)$ a function two times differentiable with continuity, such that $f(\nu_k) = y_k,\ k \in \{1, \ldots, m\}$, then, for any $a \leq \nu_1$ and $b \geq \nu_m$, it results

$$\int_a^b \left(s''(x)\right)^2 dx \leq \int_a^b \left(f''(x)\right)^2 dx$$

One consequence of this theorem is that, within all the continuous function $f(\cdot)$ with continuous first and second derivatives, the one that solves the optimization problem (2.9) is always a natural cubic splines.

Let's consider the determinations $x_1^* < x_2^* < \cdots < x_m^*$ of the variable x. If $f(\cdot)$ is a continuous function with continuous first and second derivatives, for the theorem 2.1, there exists only one natural cubic spline $s(\cdot)$ such that $s(x_k^*) = f(x_k^*),\ k \in \{1, \ldots, m\}$.

As, $s(x_i) = f(x_i)\ \forall i \in \mathcal{D}$, it results that $D(s, \boldsymbol{y}) = D(f, \boldsymbol{y})$. As $D(s, \boldsymbol{y}) = D(f, \boldsymbol{y})$ and $\int_a^b (s''(x))^2\, dx \leq \int_a^b (f''(x))^2\, dx$, it results that, for any given λ:

$$D(s, \boldsymbol{y}) + \lambda \int_a^b (s''(x))^2\, dx \ \leq\ D(f, \boldsymbol{y}) + \lambda \int_a^b (f''(x))^2\, dx$$

For this reason, if the aim of the model estimation is to minimize (2.9), for choosing f we can just consider the class of the natural cubic splines.

Other basis

In section 2.1.2 we said that natural cubic splines on knots $\nu_1, \ldots, \nu_m$ constitute a m-dimensional vector space and that a possible basis decomposition is (2.7).

Another usual basis is the *B-spline* basis. B-splines are functions defined recursively as follows.

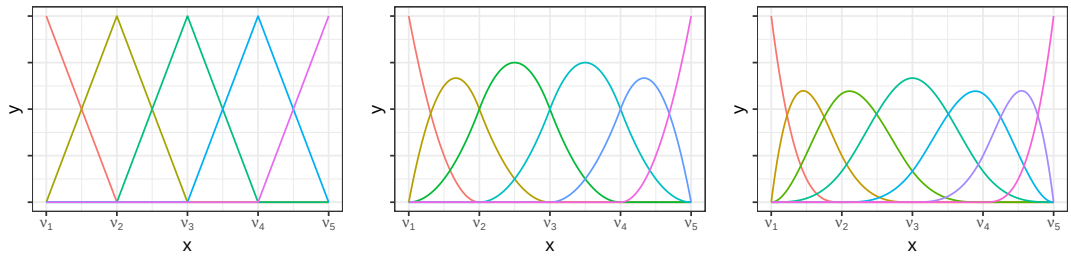
Definition 2.4 (B-splines)**.** For $k \in \{1, 2, \ldots, m-1\}$:

$$B_{0,k}(x) = \begin{cases} 1: & \nu_k < x < \nu_{k+1} \\ 0: & \text{otherwise} \end{cases}$$

For $j \geq 0$ and $k \in \{1, 2, \ldots, m+j\}$:

$$B_{j+1,k}(x) = \frac{x - \nu_{k-j-1}}{\nu_k - \nu_{k-j-1}} B_{j,k-1}(x) + \frac{\nu_{k+1} - x}{\nu_{k+1} - \nu_{k-j}} B_{j,k}(x)$$

In figure 2.9 some B-splines of different degrees are represented.



(a) degree 1 **(b)** degree 2 **(c)** degree 3

Figure 2.9: B-splines with different degrees.

It can be proven that the functions $B_{j,k}(x)$ are, given j, splines of degree j. Moreover, $B_{j,1}, \ldots, B_{j,m+j-1}$ constitute a basis of the vector space of the splines of degree j on the knots $\nu_1, \ldots, \nu_m$. In particular, $B_{3,1}, \ldots, B_{j,m+2}$ constitute a basis of the vector space of

the splines of degree 3. Therefore, if $s(x)$ is third degree spline on the knots $\nu_1, \dots, \nu_m$, it can be expressed as a linear combination of $B_{3,1}, \dots, B_{3,m+2}$:

$$s(x) = \sum_{k=1}^{m+2} \beta_k B_{3,k}(x)$$

B-splines are preferred compared to truncated polynomial as basis function, since they are less correlated and lead to more stable and less computationally expensive estimates.

GAM extensions

As in GLM, in GAM we can consider interactions between a variable with non-linear effect x_{l_2} and another variable x_{l_1}. This can be achieved by adding to the linear predictor a term such as:

$$x_{l_1} f(x_{l_2})$$

This is particularly useful when x_{l_1} is a binary variable and we want to fit two different curves for x_{l_2} in the case $x_{l_1} = 0$ and in the case $x_{l_1} = 1$.

If we want to consider a more complex interaction between two quantitative variables with non-linear effect, GAM can be extended by considering non-parametric interactions and modeling them with two-dimensional splines, such as:

$$f(x_{l_1}, x_{l_2})$$

In this case, instead of fitting a curve on (x_l, y), we will fit a flexible surface on (x_{l_1}, x_{l_2}, y). This approach can be adopted also for modeling geographical data, in which x_{l_1}, x_{l_2} are coordinates on the map, such as longitude and latitude.

Some considerations on GAM

As we have seen in this chapter, GAMs are flexible tools for fitting quantitative variables with non-linear effects.

One big advantage of GAM is that they are based on the same assumptions of GLM, except for the non-linearity of the components $f(x_l)$. The connection with GLM leads to highly interpretable results. Indeed, as we do in GLM, in GAM we can easily observe the marginal effect of a variable x_l on the response y just by plotting the graph $(x_l, f(x_l))$, while the interactions between variables are added manually so we have a full control of them.

Another big advantage is that GAM not only provides a flexible tool that produces interpretable results, but this tool works almost automatically. Indeed, while in GLM, when we have to fit a non-linear effect to a quantitative variable x_l, we have to perform a

manual process of wise splitting and polynomial fitting on the range of x_l, in GAM we do not have to explicitly specify the shape of $f(x_l)$ and everything is done by the algorithm.

This higher flexibility and automation comes at the cost of introducing more complexity in the model. Indeed, in a GAM there are much more parameters than in a GLM and this leads to a more computationally expensive fitting. Anyway, this higher complexity produces a higher machine time but significantly reduces the human time that in GLM would be needed for manual fitting.

2.1.3 Shrinkage estimators for GLM

In this section we are going to present the shrinkage estimators, that are a class of estimators particularly useful in GLM when there are many explanatory variables. All the models presented in this chapter are actually GLM. The advancements consist in the way the parameters are estimated.

The Bias-Variance Trade Off

One of the property of GLM with linear link and Normal response is that the maximum likelihood estimator $\tilde{\boldsymbol{\beta}}^{ML}$ is unbiased, that is $E\left(\tilde{\boldsymbol{\beta}}^{ML}\right) = \boldsymbol{\beta}$.[2] However, the bias is not the only relevant aspect of an estimator. If we consider the *Mean Squared Error* (MSE) of an estimator $\tilde{\beta}_j$, we get the following result:

$$MSE\left(\tilde{\beta}_j\right) \stackrel{\text{def}}{=} E\left(\left(\tilde{\beta}_j - \beta_j\right)^2\right) = \underbrace{\left(E(\tilde{\beta}_j) - \beta_j\right)^2}_{\text{Bias}^2} + \underbrace{Var\left(\tilde{\beta}_j\right)}_{\text{Variance}} \tag{2.14}$$

The idea behind shrinkage estimators is to add an amount of smart bias to $\tilde{\beta}_j$ in order to reduce its variance.

The trade-off between Bias and Variance described by equation (2.14) can be interpreted in relation to the complexity of the model. Figure 2.10 shows how the prediction error changes by increasing the complexity of the model. In linear smoothers, the complexity of the model can be measured as the effective degrees of freedom $\text{tr}(H)$ (see section 2.1.2), that in the case of the GLM with maximum likelihood estimators is just the number of parameters $p + 1$. Usually, in machine learning models, the complexity of the model is determined by tuning one or more hyper-parameters. In the case of GAM, for example, by increasing λ we obtain a less complex model, while by decreasing it we obtain a more complex model.

A model not complex enough produces estimators with low variance but high bias, that leads to underfitting. A too complex model produces estimators with low bias but high variance, that leads to overfitting.

As with maximum likelihood estimators for GLM the complexity increases with p, they are not suitable when p is large (high dimensionality).

Ridge Regression

One shrinkage estimator for GLM is the *Ridge Regression*. In Ridge Regression, the decrease in variance of $\tilde{\boldsymbol{\beta}}$ is achieved by considering the optimization problem (2.5)

[2] This property in general is not true for GLM with other links and response distributions, but it is true asymptotically: $\tilde{\boldsymbol{\beta}}^{ML} \xrightarrow{n \to +\infty} \boldsymbol{\beta}$.

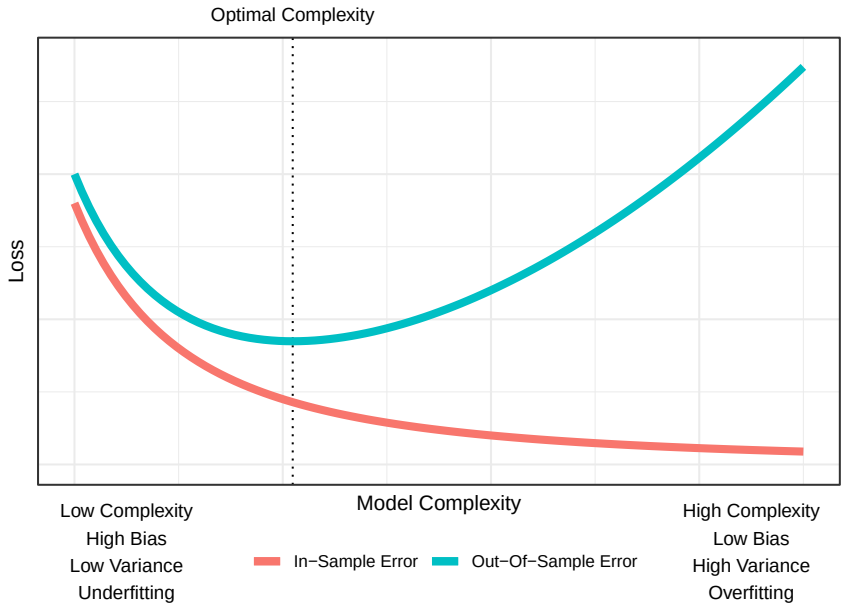


Figure 2.10: The Bias-Variance trade off. The horizontal axis shows the increase in complexity and the vertical axis shows the effect on in-sample error (training set) and out-of-sample error (test set). The error can be measured as the Mean Squared Error (MSE) or in general as the Deviance.

and adding to the Deviance a penalization term that depends on the magnitude of $\beta_1, \beta_2, \ldots, \beta_p$:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \left\{ D(\boldsymbol{\beta}, \boldsymbol{y}) + \lambda \|\boldsymbol{\beta}_{\setminus 0}\|_2^2 \right\} \tag{2.15}$$

where:

- $\boldsymbol{\beta}_{\setminus 0} = (\beta_1, \beta_2, \ldots, \beta_p)$ is the set of the GLM coefficients except for the intercept β_0;
- $\| \cdot \|_2^2$ is the L^2 norm, i.e. $\|\boldsymbol{\beta}_{\setminus 0}\|_2^2 = \sum_{j=1}^{p} \beta_j^2$;
- $\lambda \geq 0$ is an hyper-parameter that controls the penalization.

The term $\|\boldsymbol{\beta}_{\setminus 0}\|_2^2$ produces a penalization for high values of β_j. This penalization leads to a shrinkage of the coefficients. The exclusion of β_0 from the penalization term is intended to prevent the introduction of a bias towards 0 in the intercept. As the magnitude of β_j depends on the values of x_j, it is preferred to standardize all the explanatory variables to avoid distorting effects due to the unit of measure of the explanatory variables.

As for GAM, the hyper-parameter λ determines the amount of penalization given to high values of the coefficients $\hat{\beta}_j$. If $\lambda = 0$, the optimization problem (2.15) corresponds to the maximum likelihood. As λ increases, the optimal coefficients $\hat{\beta}_j$ tends to shrink

towards 0. In the limit case, if $\lambda \to +\infty$, the optimal coefficients $\hat{\beta}_j$ are all equal to 0, except for $\hat{\beta}_0$, that is equal to $g(\bar{y})$, so the estimated model corresponds to the trivial model with only the intercept $g(\mu) = \beta_0$.

An example of the effect of λ is reported in figure 2.11. The data represented has been simulated from a GLM with idenity link and Normal response. As we can see, if $\lambda = 0$, all the estimated responses $\hat{\mu}_j$ correspond to the average of the response on that group $\bar{y}_j$. As λ increases, the estimated responses move towards the global average $\bar{y}$. The speed of convergence to 0 depends on how much the group average $\bar{y}_j$ differs from the global average $\bar{y}$ and on the number of observation of the group: if in a group j there are many observations, we have a lot of information on that group, the group average $\bar{y}_j$ have a small variance and it is a reliable estimate for μ_j, while if there are few observations, the group average $\bar{y}_j$ have a high variance and it is not very reliable.

The optimal point for $\lambda = 0$ can be obtained through a Cross Validation as seen in section 2.1.2.

The case of GLM with identity link and Normal distribution is particularly convenient for the interpretation, because, in this case, the optimization problem (2.15) has an explicit solution. To make the results more interpretable, let's assume that the response variables have $E(Y) = 0$ and that the explanatory variables are centered on 0, i.e. $\bar{x}_j = 0,\ j \in \{1, 2, \ldots, p\}$. That means that $\beta_0 = 0$ and the model is $E(Y) = \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p$. With these assumptions, we obtain:

$$\hat{\boldsymbol{\beta}}_{\lambda,\setminus 0} = \left(\boldsymbol{X}^t \boldsymbol{X} + \lambda I_p\right)^{-1} \boldsymbol{X}^t y \tag{2.16}$$

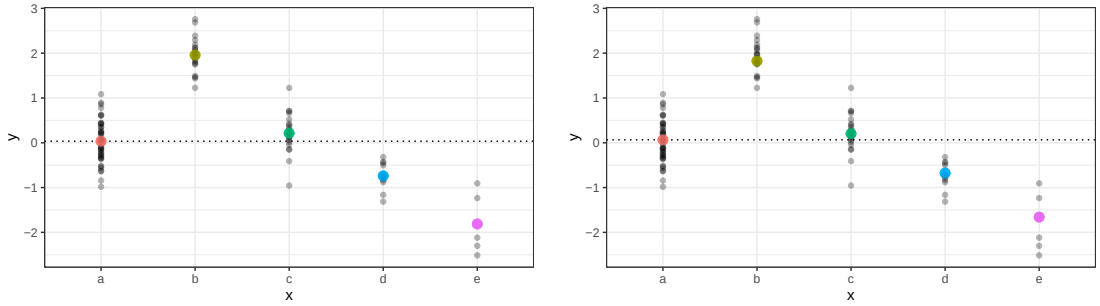
where:

- $\hat{\boldsymbol{\beta}}_{\lambda,\setminus 0} = \left(\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p\right)$;
- $\boldsymbol{X}$ is the design matrix without the first column of 1s, as the intercept is excluded from the model;
- I_p is the identity matrix with dimension p.

From the formula (2.16) we can see that, while the maximum likelihood estimator is unbiased, if $\lambda > 0$ the Ridge estimator is biased.

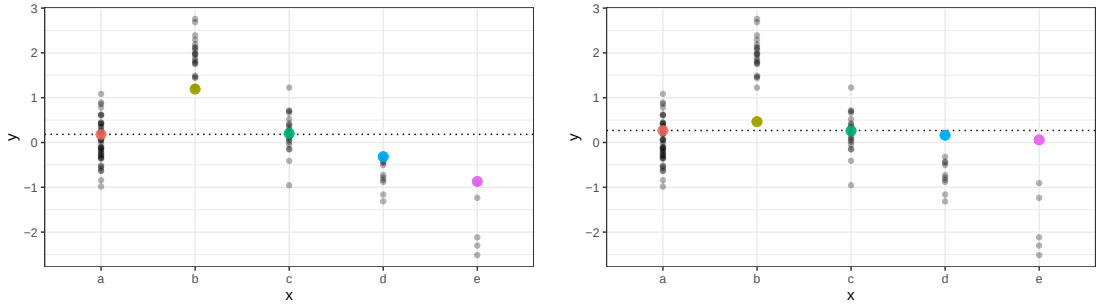
$$\begin{aligned} E\left(\tilde{\boldsymbol{\beta}}_{0,\setminus 0}\right) &= \left(\boldsymbol{X}^t \boldsymbol{X}\right)^{-1} \boldsymbol{X}^t E(Y) \\ &= \left(\boldsymbol{X}^t \boldsymbol{X}\right)^{-1} \boldsymbol{X}^t \boldsymbol{X} \boldsymbol{\beta}_{\setminus 0} \\ &= \boldsymbol{\beta}_{\setminus 0} \end{aligned}$$

$$\begin{aligned} E\left(\tilde{\boldsymbol{\beta}}_{\lambda,\setminus 0}\right) &= \left(\boldsymbol{X}^t \boldsymbol{X} + \lambda I_p\right)^{-1} \boldsymbol{X}^t E(Y) \\ &= \left(\boldsymbol{X}^t \boldsymbol{X} + \lambda I_p\right)^{-1} \boldsymbol{X}^t \boldsymbol{X} \boldsymbol{\beta}_{\setminus 0} \\ &\neq \boldsymbol{\beta}_{\setminus 0} \end{aligned}$$



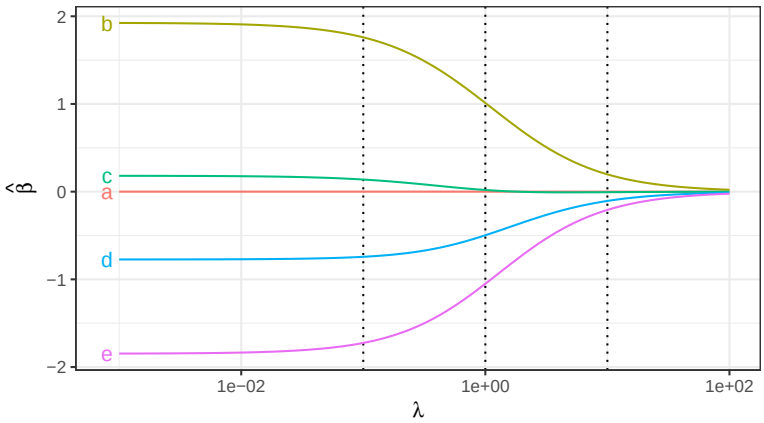
(a) $\lambda = 0$

(b) $\lambda = 0.1$



(c) $\lambda = 1$

(d) $\lambda = 10$



(e) $\hat{\beta}_j$ for different values of λ

Figure 2.11: Ridge Regression coefficients for different levels of the penalization parameter λ. If $\lambda = 0$, the coefficients correspond to the maximum likelihood coefficients. As λ increases, the coefficients are shrunk towards 0.

Moreover, from the formula (2.16) we find that, even if there is multicollinearity and $\boldsymbol{X}^t\boldsymbol{X}$ is not invertible, the Ridge estimator is computable. This aspect is particularly interesting when $p > n$.

If we assume that the explanatory variables are independent and standardized, i.e. $\boldsymbol{X}^t\boldsymbol{X} = I_p$, we can further simplify the expression (2.16) to:

$$\hat{\boldsymbol{\beta}}_{\lambda,\setminus 0} = \frac{1}{1+\lambda}\boldsymbol{X}^t y = \frac{1}{1+\lambda}\hat{\boldsymbol{\beta}}_{0,\setminus 0}$$

that results in:

$$\begin{aligned} E\left(\tilde{\boldsymbol{\beta}}_{\lambda,\setminus 0}\right) &= \frac{1}{1+\lambda}\boldsymbol{\beta}_{\setminus 0} \\ Var\left(\tilde{\boldsymbol{\beta}}_{\lambda,\setminus 0}\right) &= \frac{1}{(1+\lambda)^2}Var\left(\tilde{\boldsymbol{\beta}}_{0,\setminus 0}\right) \end{aligned}$$

This result means that, if the explanatory variables are independent, the Ridge penalization shrinks all the estimated coefficients by a factor of $\frac{1}{1+\lambda}$ and reduce their variance by a factor of $\frac{1}{(1+\lambda)^2}$.

From formula (2.16) we also find that, as in GAM, if the link is identity and the response is Normal, the Ridge regression is a linear smoother. In this case the smoothing matrix is:

$$H_\lambda = \boldsymbol{X}\left(\boldsymbol{X}^t\boldsymbol{X} + \lambda I_p\right)^{-1}\boldsymbol{X}^t$$

In the case of independent explanatory variables we get:

$$H_\lambda = \frac{1}{1+\lambda}\boldsymbol{X}\boldsymbol{X}^t = \frac{1}{1+\lambda}H_0$$

and then $\text{tr}\left(H_\lambda\right) = \frac{p}{1+\lambda}$. This result means that, by increasing λ, the effective number of degrees of fredom decreases.

LASSO Regression

Another shrinkage estimator for GLM is the Least Absolute Shrinkage and Selection Operator (LASSO). LASSO is based on the same idea of Ridge Regression, but, instead of considering a penalization based on L^2 norm, it considers a penalization based on L^1 norm:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min}\left\{D(\boldsymbol{\beta}, \boldsymbol{y}) + \lambda\|\boldsymbol{\beta}_{\setminus 0}\|_1\right\} \tag{2.17}$$

where:

- $\boldsymbol{\beta}_{\setminus 0} = (\beta_1, \beta_2, \ldots, \beta_p)$ is the set of the GLM coefficients except for the intercept β_0;
- $\|\cdot\|_1$ is the L^1 norm, i.e. $\|\boldsymbol{\beta}_{\setminus 0}\|_1 = \sum_{j=1}^{p} |\beta_j|$;
- $\lambda \geq 0$ is an hyper-parameter that controls the penalization.

An example of the effect of the L^1 penalization for different values of λ is shown in figure 2.12. The simulated dataset is the same used for the Ridge example in figure 2.11. As we can see from the plots, the substantial difference between Ridge and LASSO is that in LASSO from a certain value of λ the coefficients are shrunk exactly to 0. While in Ridge that is just a limit property, in LASSO, for each coefficient β_j there is a level of the penalization parameter λ'_j such that for $\lambda \geq \lambda'_j$, the estimated coefficient $\hat{\beta}_j$ is forced to exactly 0.

This property of the LASSO Regression can be derived from a different representation of the optimization problems (2.15) and (2.17). It can be proven that in general, considering a penalization given by the L^d norm, the unconstrained optimization problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \left\{ D(\boldsymbol{\beta}, \boldsymbol{y}) + \lambda \|\boldsymbol{\beta}_{\setminus 0}\|_d^d \right\} \tag{2.18}$$

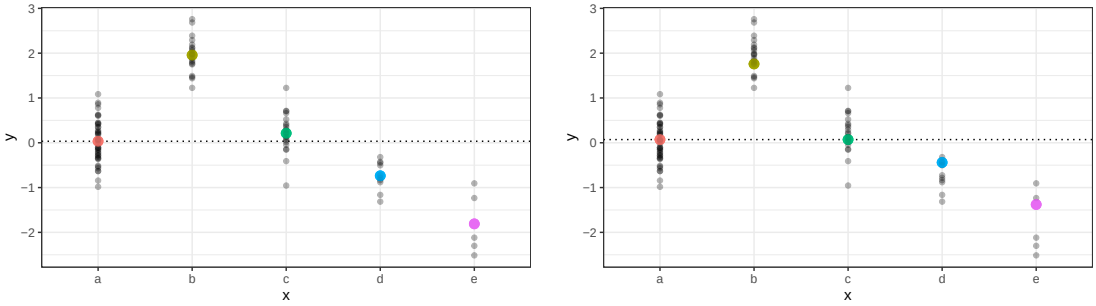
is equivalent to the constrained optimization problem:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}:\ \|\boldsymbol{\beta}_{\setminus 0}\|_d^d \leq s_\lambda}{\arg\min} D(\boldsymbol{\beta}, \boldsymbol{y}) \tag{2.19}$$

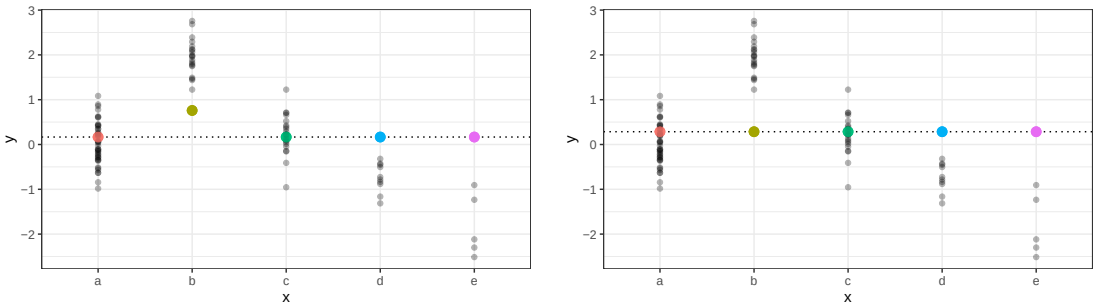
where s_λ is a quantity that depends on λ.

The representation (2.19) provides a useful geometric interpretation of the optimization problem. Figure 2.13 shows the visual representation of the optimization problem (2.19) in the Ridge case ($d = 2$) and in the LASSO case ($d = 1$). The axes represent the component of $\boldsymbol{\beta}$. The point $\hat{\boldsymbol{\beta}}^{ML}$ represents the maximum likelihood estimator for $\boldsymbol{\beta}$, that is the optimal point for the Deviance $D(\boldsymbol{\beta}, \boldsymbol{y})$ without any constraints. The ellipses around $\hat{\boldsymbol{\beta}}^{ML}$ represent the contour lines of $D(\boldsymbol{\beta}, \boldsymbol{y})$. In the Normal case with identity link they are concentric ellipses centered in $\hat{\boldsymbol{\beta}}^{ML}$. The grey area around the axes intersection represents the feasibility region determined by $\|\boldsymbol{\beta}\|_d^d \leq s_\lambda$. This area, in the Ridge case corresponds to the circle $\|\boldsymbol{\beta}\|_2^2 \leq s_\lambda$, while in the LASSO case corresponds to the square $\|\boldsymbol{\beta}\|_1^1 \leq s_\lambda$. $\hat{\boldsymbol{\beta}}^{Ridge}$ and $\hat{\boldsymbol{\beta}}^{LASSO}$ are respectively the optimal point conditioned to the Ridge constraint and the optimal point conditioned to the LASSO constraint. The sharpness of the LASSO feasibility region implies that the optimal point $\hat{\boldsymbol{\beta}}^{LASSO}$ could fall into one of the corners of the square leading one of the coefficients $\hat{\beta}_j^{LASSO}$ to be exactly equal to 0. In general, if there are more than two explanatory variables, the LASSO feasibility region is an hyper-cube and the LASSO attains solutions with many coefficients exactly equal to 0.

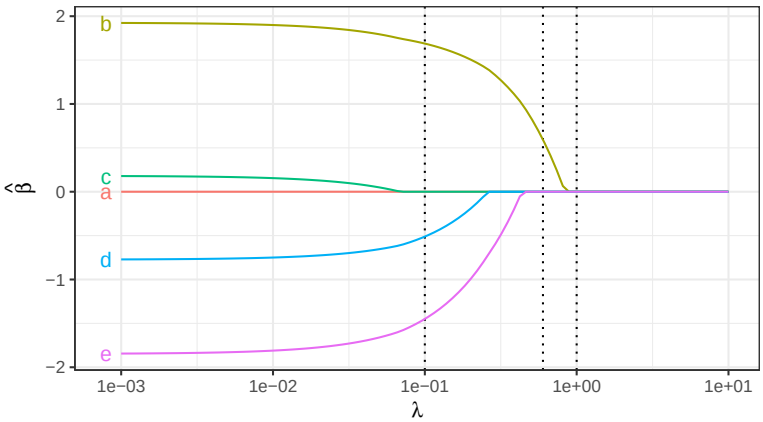
The fact that in LASSO Regression the estimated coefficients can be exactly equal to 0 is a precious benefit. Indeed, LASSO Regression performs a feature selection removing



(a) $\lambda = 0$ **(b)** $\lambda = 0.1$

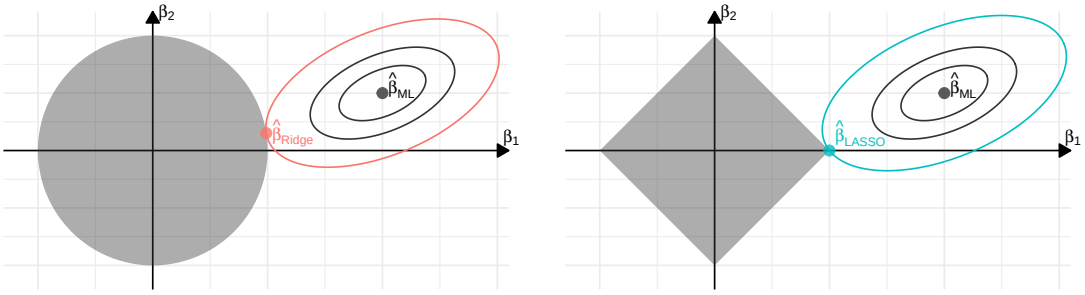


(c) $\lambda = 0.6$ **(d)** $\lambda = 1$



(e) $\hat{\beta}_j$ for different values of λ

Figure 2.12: LASSO Regression coefficients for different levels of the penalization parameter λ. If $\lambda = 0$, the coefficients correspond to the maximum likelihood coefficients. As λ increases, the coefficients are shrunk towards 0. High values of λ force the coefficients to be exactly equal to 0.



(a) Ridge **(b)** LASSO

Figure 2.13: Geometrical interpretation of the optimization problem for Ridge and LASSO. The sharpness of the LASSO feasibility region implies that the optimal point $\hat{\boldsymbol{\beta}}^{LASSO}$ could fall into one of the corners of the square leading one of the coefficients $\hat{\beta}_j^{LASSO}$ to be exactly equal to 0.

the variables that are not relevant for predicting the response. The LASSO fitting is much more efficient than the other procedures we have seen for feature selection in GLM such as the stepwise selection based on AIC or other criteria (section 2.1.1) and it is better scalable for datasets with many variables. It is also possible to use the variables selected by the LASSO regression and giving them as inputs for a maximum likelihood fitting. If in the dataset there are many variables but only few of them are relevant for the response, this procedure can return better predictions than the LASSO regression by itself.

Elastic Net

Compared to the Ridge Regression, the LASSO Regression has the benefit of performing a feature selection by forcing many coefficients to 0. Anyway this doesn't necessarily mean that LASSO estimates always outperform Ridge estimates. It strongly depends on the case. In general, if in a dataset only few of the explanatory variables have an effect on the response, the LASSO will produce better estimates, but, if all the variables bring a small amount of information on the response, then the LASSO will suppress some of this information, while the Ridge will catch it. The problem is that in practice, when we have a real dataset, we do not know in which case we are. The *Elastic Net* is a generalization of the Ridge and the LASSO that provides a solution for this kind of problem.

The Elastic Net consists in a penalized Deviance optimization problem in which the penalization term is a mixture of the Ridge penalization and the LASSO penalization:

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min} \left\{ D(\boldsymbol{\beta}, \boldsymbol{y}) + \lambda \sum_{j=1}^{p} \left(\alpha|\beta_j| + (1-\alpha)|\beta_j|^2 \right) \right\} \tag{2.20}$$

where $\alpha \in [0, 1]$ is a hyper-parameter that weighs the two penalization components.

Looking to the equation (2.20) it is clear that, if $\alpha = 0$, the Elastic Net corresponds to the Ridge Regression, while, if $\alpha = 1$, the Elastic Net corresponds to the LASSO regression. If $\alpha \in]0, 1[$, the result will be a compromise between the two.

The hyper-parameter α can be estimated together with λ in a Cross Validation procedure (see section 2.1.1). If the data suggests that many variables are useful for predicting the response, the estimated hyper-parameter $\hat{\alpha}$ will be close to 0, while if the data suggests that only few variables are useful, the estimated hyper-parameter $\hat{\alpha}$ will be close to 1.

Some considerations on Shrinkage Estimators

As we have said, shrinkage estimators are particularly useful when the number of parameters p in the model is large (high dimensionality) and the maximum likelihood estimator have a high variance. This could happen when we have some qualitative variables with many modalities, as for example the make and the model of the insured vehicle in car insurance. As we have seen with the examples represented in figure 2.11 and 2.12, the penalization will shrink more the coefficients in the groups with few observations and only the groups with an average response $\bar{y}_j$ significantly different from the global average response $\bar{y}$ will emerge. This procedure is more efficient and more robust to overfitting than grouping modalities based on hypothesis testing or other criteria.

If we are in a case in which we already performed a satisfying feature selection and we only want to shrink some of the coefficients we can apply the penalization only to them. For example, in the case of the variables make and model, we can consider an Elastic Net penalization only to the coefficients β_j corresponding to the modalities of those variables. This technique is useful also if we have explanatory variables with distribution highly unbalanced. For example, if we consider the variable "number of claims experienced in the previous year", we will have most of the observation in the modality corresponding to 0 claims, very few in the modality corresponding to 1 claim and almost nobody with 2 or more claims. If we are fitting a model for the claim frequency and we consider the variable x_j that indicates whether the policyholder experienced one or more claims in the previous year $x_j = 1$ or not $x_j = 0$, it is likely that the maximum likelihood estimator for the coefficient $\hat{\beta}_{x_j=1}$ will be greater than 0, as the policyholders that experienced claims are usually more inclined to experience more of them in the future. However, given that there are just few observation with $x_j = 1$ it is possible that the coefficient $\hat{\beta}_{x_j=1}$ is not significantly different from 0. If our only tool is the maximum likelihood estimator, we have to choose whether to insert the variable x_j in the model or not. If we don't consider it we are probably discarding some potentially useful information, while if we consider it and we estimate its coefficient with maximum likelihood we risk overfitting the data. With shrinkage estimators we can choose to insert the variable x_j in the model and fitting it with a penalization. This way we exploit that information but we prevent overfitting.

Another interesting observation about GLM fitting is that in the practice, when

maximum likelihood estimates are adopted, what is done is not just fitting a model with all the variables, but a feature selection is conducted. The fact that a variable x_j is inserted in the model depends on whether the feature selection procedure selects that variable or not. That corresponds to estimating the coefficient β_j with an estimator $\tilde{\beta}_j$ that is equal to the maximum likelihood estimator $\tilde{\beta}_j^{ML}$ when the coefficient pass a specific criterion and is equal to 0 when $\tilde{\beta}_j^{ML}$ doesn't pass that criterion. This criterion could be something objective as for example the decrease in AIC and the significance in a hypothesis testing on $\tilde{\beta}_j^{ML}$, and also something more subjective based also on the domain knowledge on the person that is conducting the modeling. Actually, this feature selection procedure introduces a bias towards 0 of the coefficient $\tilde{\beta}_j$ that reduces its variance as all the coefficients not relevant for prediction are set equal to 0 and only the relevant ones are fitted with maximum likelihood. If we consider a binary variable x_j, in all the procedures commonly used for feature selection, the probability that $\tilde{\beta}_j$ passes the procedure or not depends on how strong the effect β_j is and how many observations there are in the classes $x_j = 0$ and $x_j = 1$.

We also mention that the Shrinkage Estimators can be used in synergy with GAMs. Indeed it is possible to fit to the quantitative variables a cubic spline with a GAM penalization based on the second derivative of the spline and to the qualitative variables a shrinkage estimator with an Elastic Net penalization.

$$\hat{\boldsymbol{f}} = \arg\min_{\boldsymbol{f}} \left\{ D(\boldsymbol{f}, \boldsymbol{y}) + \sum_{l=1}^{q} \lambda_l \int_{a_l}^{b_l} \left(f_l''(x_l)\right)^2 dx + \lambda_{EN} \sum_{j=1}^{p} \left(\alpha|\beta_j| + (1-\alpha)|\beta_j|^2\right) \right\} \tag{2.21}$$

where the coefficients β_j considered in the Elastic Net penalization are only the ones corresponding to qualitatives variables.

2.1.4 Bayesian GLM

In this section we are going to present the Bayesian estimators for GLM. The novelty compared to Maximum Likelihood estimators consists in the fact that Bayesian statistics introduces the idea of prior information that, used in inference, brings a bias to the estimates. As we will see in section 2.1.4, Ridge Regression and LASSO Regression can be interpreted as Bayesian estimators.

The Bayesian framework

In classical inference what is commonly done for estimating an unknown parameter θ using an observed sample $\boldsymbol{y} = (y_1, y_2, \dots, y_n)$, is to consider the value $\hat{\theta}^{ML}$ that maximizes the likelihood:

$$\hat{\theta}^{ML} = \arg\max_{\theta \in \Theta} L(\theta) = \arg\max_{\theta \in \Theta} p(\boldsymbol{y}|\theta)$$

where $p(\boldsymbol{y}|\theta)$ is the probability (or the density) of the sample $\boldsymbol{y}$ given the parameter θ. That means that, within all the possible parameters $\theta \in \Theta$, we select the most likely one, that is the one that, conditional to it, returns the maximum probability for the observed sample $\boldsymbol{y}$. We highlight that $L(\theta)$ is not a probability distribution on θ, so for example, in general, $\int_\Theta L(\theta) d\theta \neq 1$.

Bayesian Inference introduces the concept of prior distribution of the parameter: $\pi(\theta)$. This distribution represents how probable we assume the different values of $\theta \in \Theta$ are, prior to the observation of the sample. In classical statistics, the parameter θ is seen as a specific real number that is unknown. In the Bayesian framework, the parameter θ is seen as a random variable and its distribution represents our information on it. This aspect introduces a subjective point of view of probability.

With the *Bayes Theorem*, having a prior distribution $\pi(\theta)$, we can compute the posterior distribution for θ given the observed sample $\boldsymbol{y}$:

$$\pi(\theta|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\theta)\pi(\theta)}{p(\boldsymbol{y})}$$

$\pi(\theta|\boldsymbol{y})$ is actually a probability distribution, so we can compute probabilities and make predictions on θ given the distribution $\pi(\theta|\boldsymbol{y})$.

Two useful statistics based on the posterior distribution $\pi(\theta|\boldsymbol{y})$ are the expected value of θ given $\boldsymbol{y}$:

$$E(\theta|\boldsymbol{y}) = \int_\Theta \theta \pi(\theta|\boldsymbol{y}) d\theta$$

and the mode of θ given $\boldsymbol{y}$:

$$\text{Mode}(\theta|\boldsymbol{y}) = \arg\max_{\theta \in \Theta} \pi(\theta|\boldsymbol{y})$$

These two statistics can be used as point estimates for the parameter θ. In the following we are going to refer to $\text{Mode}(\theta|\boldsymbol{y})$ as the *Maximum a Posteriori* (MAP) estimate:

$$\hat{\theta}^{MAP} = \text{Mode}(\theta|\boldsymbol{y}) = \arg\max_{\theta\in\Theta} \pi(\theta|\boldsymbol{y})$$

Bayesian estimator for the mean of a Normal distribution

One example to dive into the the Bayesian inference that is useful to better understand the logic of Bayesian estimators in GLM is the inference on the mean of a Normal distribution from an observed sample.

Let's assume we have a independent and identically distributed sample $\boldsymbol{y} = (y_1, y_2, \ldots, y_n)$ from a Normal distribution:

$$p(y_i|\mu, \sigma^2) \sim \mathcal{N}(\mu, \sigma^2),\ i \in \{1, 2, \ldots, n\} \tag{2.22}$$

Let's assume that σ^2 is known and we want to infer the value of μ. In the Bayesian framework, μ is a random variable, so the first thing we have to do is to define a prior distribution for it. Let's assume:

$$\pi(\mu) \sim \mathcal{N}(\mu_0, \sigma_0) \tag{2.23}$$

where μ_0 and σ_0 are known parameters.

Under these assumptions we can compute the posterior distribution as:

$$\pi(\mu|\boldsymbol{y}) = \frac{p(\boldsymbol{y}|\mu)\pi(\mu)}{p(\boldsymbol{y})}$$

As we want to conduct inference on the parameter μ and the denominator $p(\boldsymbol{y})$ does not depend on μ, we can just consider the numerator:

$$\pi(\mu|\boldsymbol{y}) \propto p(\boldsymbol{y}|\mu)\pi(\mu) \tag{2.24}$$

It is possible to prove that, by substituting in formula (2.24) the likelihood $p(\boldsymbol{y}|\mu)$ and prior $\pi(\mu)$ with (2.22) and (2.23), we get:

$$\pi(\mu|\boldsymbol{y}) \sim \mathcal{N}\left(\mu_n, \sigma_n\right) \tag{2.25}$$

where:

$$\mu_n = \frac{\frac{n}{\sigma^2}\bar{y} + \frac{1}{\sigma_0^2}\mu_0}{\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}} \tag{2.26}$$

$$\sigma_n^2 = \left(\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}\right)^{-1} \tag{2.27}$$

The posterior distribution for the mean $\pi(\mu|\boldsymbol{y})$ is still a Normal distribution with parameters (μ_n, σ_n^2) that depend on the initial parameters (μ_0, σ_0^2), the observed sample average $\bar{y}$ and the sample size n.

Figure 2.14 shows the prior $\pi(\mu)$, the likelihood $p(\boldsymbol{y}|\mu)$ and the posterior distribution $\pi(\mu|\boldsymbol{y})$ for different values of the initial parameters and for different sample sizes n. As we can see, $\pi(\mu|\boldsymbol{y})$ lies between $\pi(\mu)$ and $p(\boldsymbol{y}|\mu)$.

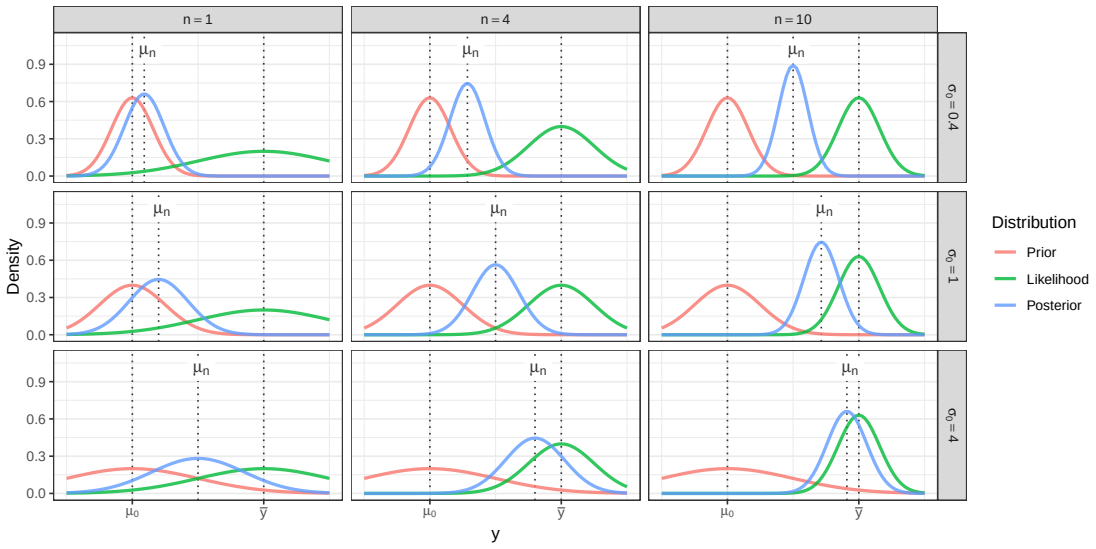


Figure 2.14: Prior $\pi(\mu)$, likelihood $p(\boldsymbol{y}|\mu)$ and posterior distribution $\pi(\mu|\boldsymbol{y})$ for the estimate of the mean from a Normal distribution. The panels shows $\pi(\mu)$, $p(\boldsymbol{y}|\mu)$ and $\pi(\mu|\boldsymbol{y})$ for different values of the prior variance σ_0 (rows) and different sample sizes n (columns).

As $\pi(\mu|\boldsymbol{y})$ is a Normal distribution, the expected value $E(\mu|\boldsymbol{y})$ and the mode $\text{Mode}(\mu|\boldsymbol{y})$ coincide, so our Maximum a Posteriori estimate for μ will be:

$$\hat{\mu}^{MAP} = \text{Mode}(\mu|\boldsymbol{y}) = E(\mu|\boldsymbol{y}) = \mu_n = \frac{\frac{n}{\sigma^2}\bar{y} + \frac{1}{\sigma_0^2}\mu_0}{\frac{n}{\sigma^2} + \frac{1}{\sigma_0^2}}$$

From the formula (2.26) we get some interesting insights on $E(\mu|\boldsymbol{y}) = \mu_n$. μ_n is a weighted average between the prior mean μ_0 and the sample mean $\bar{y}$, so it stays between μ_0 and $\bar{y}$. The weight for μ_0 and $\bar{y}$ are given respectively by the reciprocal of the prior variance $\frac{1}{Var(\mu)} = \frac{1}{\sigma_0^2}$ and the reciprocal of the sample mean variance $\frac{1}{Var(\bar{y})} = \frac{n}{\sigma^2}$. That means that the lower the prior variance is, the greater the weight for μ_0 is, and the lower the sample mean variance is, the greater the weight for $\bar{y}$ is. The reciprocal of the variance can be interpreted as the amount of information we have on that estimate. Thus, a prior distribution with a lower variance is a more informative prior, while a prior distribution with an higher variance is a less informative prior. The sample mean variance $Var(\bar{y}) = \frac{\sigma^2}{n}$ depends on σ^2 and n. If n increases, $Var(\bar{y})$ decreases and so

the weight for $\bar{y}$ increases. That corresponds to saying that, by increasing the sample size n, we give more credibility to the observed sample mean $\bar{y}$.

From the formula (2.27) we see that the reciprocal of $Var(\mu|\boldsymbol{y}) = \sigma_n^2$ is the sum between the reciprocal of $Var(\mu) = \sigma_0^2$ and the reciprocal of $Var(\bar{y}) = \frac{\sigma^2}{n}$. Interpreting the reciprocal of the variance as the *information* on that estimate, this result on $Var(\mu|\boldsymbol{y})$ corresponds to saying that the information on the posterior distribution is the sum of the prior information and the information obtained by the sample.

On the limit case, if $n \to +\infty$, the posterior mean converges to the observed sample mean $\mu_n \to \bar{y}$ and the posterior variance converges to zero $\sigma_n \to 0$.

Another limit case is to consider a prior distribution with $\sigma_0^2 = +\infty$, that is $\pi(\mu) \propto k,\ k \in]0, +\infty[$. This is an improper prior, as for any value of $k \in]0, +\infty[$, we get $\int_{\mathbb{R}} \pi(\mu) d\mu = +\infty$. This kind of prior distribution is called *Non-informative prior*, as it gives no information on the parameter. In this case, by (2.26) and (2.27) we get:

$$\mu_n = \bar{y}$$
$$\sigma_n^2 = \frac{\sigma^2}{n}$$

From equation, (2.24) we get that, if we use a non-informative prior, the posterior distribution is proportional to the likelihood $\pi(\mu|\boldsymbol{y}) \propto p(\boldsymbol{y}|\mu)$. That means that the Maximum a Posteriori estimates $\hat{\mu}^{MAP}$ corresponds to the maximum likelihood estimates $\hat{\mu}^{ML}$. This results gives a new interpretation of the Maximum Likelihood estimates, as it can be always seen as a Bayesian posterior estimate with a Non-informative prior distribution.

In this example, for simplicity, we considered σ^2 as a known parameter. In practice this isn't a common situation. Usually σ^2 have to be estimated like μ. In the Bayesian framework we must assign a prior distribution $\pi(\sigma^2)$ to it. If we don't want to introduce a bias, we can choose a non informative prior.

The problem of inference on the mean μ of a Normal distribution $y_i \sim \mathcal{N}(\mu, \sigma^2)$ with a Normal prior assigned to $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$ is particularly convenient because it produces an explicit solution for $\pi(\mu|\boldsymbol{y})$ that it is easily interpretable and can be easily computed. Given the sample distribution $p(\boldsymbol{y}|\theta)$, the prior distributions $\pi(\theta)$ that returns a posterior distribution $\pi(\theta|\boldsymbol{y})$ from the same family of the prior are called *conjugate prior*. Other examples of conjugate priors are the Beta distribution for the probability of success p in a Binomial sample and the Gamma distribution for the parameter λ in a Poisson sample.

In general we are not constrained to use conjugate priors and we can choose for $\pi(\theta)$ the distribution that better describes our prior information. When there is not an explicit solution for $\pi(\theta|\boldsymbol{y})$, it can be computed numerically with simulation techniques. The techniques commonly adopted are based on *Markov Chain Monte Carlo* (MCMC). These are highly flexible techniques, but they come with a high computational cost.

Bayesian estimators for GLM

The Bayesian approach can be applied to GLM by adopting prior distribution on the coefficients $(\boldsymbol{\beta}, \phi)$. By assuming a prior distribution on $\boldsymbol{\beta}$ we will introduce a bias driven by our prior information.

In section 2.1.1 we saw that in GLM the common estimator used for $\boldsymbol{\beta}$ is the Maximum Likelihood estimator $\tilde{\boldsymbol{\beta}}^{ML}$ with determinations:

$$\hat{\boldsymbol{\beta}}^{ML} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\max}\, L\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right) \tag{2.28}$$

If in equation (2.28) we substitute $L\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right)$ with $\pi\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right)$, we obtain the Maximum a Posteriori estimator $\tilde{\boldsymbol{\beta}}^{MAP}$

$$\hat{\boldsymbol{\beta}}^{MAP} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\max}\, \pi\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right) = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\max} \left\{L\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right) \pi(\boldsymbol{\beta}, \phi)\right\} \tag{2.29}$$

If we assume a non informative prior distribution for ϕ, the equation becomes:

$$\hat{\boldsymbol{\beta}}^{MAP} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\max} \left\{L\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right) \pi(\boldsymbol{\beta})\right\} \tag{2.30}$$

Considering the log-likelihood $\ell\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right) = \log\left(L\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right)\right)$, the optimization problem (2.30) becomes:

$$\hat{\boldsymbol{\beta}}^{MAP} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\max} \left\{\ell\left(\boldsymbol{\beta}, \phi \mid \boldsymbol{y}\right) + \log\left(\pi(\boldsymbol{\beta})\right)\right\} \tag{2.31}$$

And expressed in terms of deviance $D(\hat{\boldsymbol{\beta}}, \boldsymbol{y}) = -2\phi \left(\ell\left(\hat{\boldsymbol{\beta}}, \phi; \boldsymbol{y}\right) - \ell_S\left(\boldsymbol{\beta}^*, \phi; \boldsymbol{y}\right)\right)$:

$$\hat{\boldsymbol{\beta}}^{MAP} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{p+1}}{\arg\min} \left\{D(\boldsymbol{\beta}, \boldsymbol{y}) - 2\phi \log\left(\pi(\boldsymbol{\beta})\right)\right\} \tag{2.32}$$

From equation (2.32) we find that, adding a prior distribution $\pi(\boldsymbol{\beta})$, corresponds to adding a term $-2\phi \log\left(\pi(\boldsymbol{\beta})\right)$ to the deviance in the optimization problem that defines the estimator. In particular, if the prior is non-informative, we get that $\pi(\boldsymbol{\beta}) \propto k,\ k \in]0, +\infty[$ does not depend on $\boldsymbol{\beta}$, so the optimization problem corresponds to the Maximum Likelihood.

Ridge and LASSO Regression as Bayesian estimators for GLM

Let's assume the parameters $\beta_1, \beta_2, \ldots, \beta_p$ to be identically distributed with *Normal* distribution centered in 0:

$$\pi(\beta_j) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2\sigma^2}\beta_j^2}, \quad j \in \{1, 2, \ldots, p\}$$

and let's assign to β_0 a non informative distribution:

$$\pi(\beta_0) \propto k,\ k \in]0, +\infty[$$

If we also assume that the parameters are independent, we get:

$$\pi(\boldsymbol{\beta}) \propto \prod_{j=1}^{p} \pi(\beta_j) \propto e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{p}\beta_j^2} \tag{2.33}$$

By substituting (2.33) to (2.32), we get:

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{MAP} &= \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min} \left\{ D(\boldsymbol{\beta}, \boldsymbol{y}) - 2\phi \log\left(e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{p}\beta_j^2} \right) \right\} \\ &= \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min} \left\{ D(\boldsymbol{\beta}, \boldsymbol{y}) + \frac{\phi}{\sigma^2}\sum_{j=1}^{p}\beta_j^2 \right\} \end{aligned} \tag{2.34}$$

From the expression (2.34), if we substitute $\frac{\phi}{\sigma^2}$ with λ, we obtain the optimization problem of the Ridge regression (2.15). That means that the Ridge estimator can be seen as a Maximum a Posteriori estimator in which the prior distribution for $\beta_1, \beta_2, \ldots, \beta_p$ are independent Normal centered in 0 with the same variance σ^2. From the substitution $\lambda = \frac{\phi}{\sigma^2}$ we can also interpret the role of λ and σ^2. A lower σ^2 corresponds to a more informative prior that gives more credibility to the prior mean $E(\beta_j) = 0$. In the Ridge usual parametrization, this corresponds to a greater λ and so an higher weight to the penalization $\sum_{j=1}^{p}\beta_j^2$, that brings a higher shrinkage to the estimates.

With the same approach, we can assume $\beta_1, \beta_2, \ldots, \beta_p$ to be identically distributed with *Laplace* distribution centered in 0:

$$\pi(\beta_j) = \frac{1}{2b} e^{-\frac{|\beta_j|}{b}}, \quad j \in \{1, 2, \ldots, p\}$$

Under this assumption, the prior distribution for the coefficients becomes:

$$\pi(\boldsymbol{\beta}) \propto \prod_{j=1}^{p} \pi(\beta_j) \propto e^{-\frac{1}{b}\sum_{j=1}^{p}|\beta_j|} \tag{2.35}$$

And the optimization problem (2.32) becomes:

$$\begin{aligned}\hat{\boldsymbol{\beta}}^{MAP} &= \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min} \left\{ D(\boldsymbol{\beta}, \boldsymbol{y}) - 2\phi \log\left(e^{-\frac{1}{b}\sum_{j=1}^{p}|\beta_j|} \right) \right\} \\ &= \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min} \left\{ D(\boldsymbol{\beta}, \boldsymbol{y}) + \frac{2\phi}{b}\sum_{j=1}^{p}|\beta_j| \right\} \end{aligned} \tag{2.36}$$

By substituting $\frac{2\phi}{b}$ with λ in equation (2.36), we obtain the optimization problem of the LASSO regression (2.17). In the Laplace distribution the variance is $Var(\beta_j) = 2b^2$.

By decreasing b, the variance in $\pi(\beta_j)$ decreases, so the prior is more informative and we give more credibility to the prior mean 0. As for Ridge regression, a lower variance in the prior distribution translates into an higher penalization parameter λ and a higher shrinkage for the coefficients.

With the same approach, if we assume the following prior distribution:

$$\pi(\beta_j) \propto e^{-\frac{1}{k}\left(\alpha|\beta_j|+(1-\alpha)\beta_j^2\right)}, \quad \alpha \in [0,1],\ k \in]0,+\infty[,\ j \in \{1,2,\dots,p\}$$

we obtain the Elastic Net optimization problem:

$$\hat{\boldsymbol{\beta}}^{MAP} = \underset{\boldsymbol{\beta}\in\mathbb{R}^{p+1}}{\arg\min} \left\{ D(\boldsymbol{\beta}, \boldsymbol{y}) + \frac{2\phi}{k} \sum_{j=1}^{p} \left(\alpha|\beta_j| + (1-\alpha)\beta_j^2\right) \right\}$$

where $k > 0$ is a constant that determine the variance of the prior distribution.

Figure 2.15 shows a Normal, a Laplace and a Elastic Net prior distribution with $\alpha = \frac{1}{2}$. All the distributions represented have unit variance. As we can see from the plot, the Laplace distribution has a peak on its mean. This peak is the responsible of forcing to exactly 0 the unimportant coefficients in the LASSO regression. The Elastic Net prior is a mixture between the Normal distribution and the Laplace distribution.

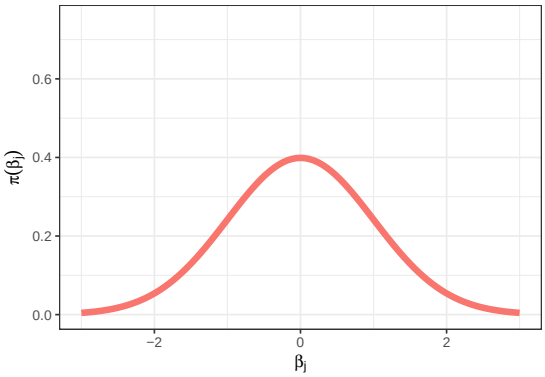
The approach usually adopted in Ridge and LASSO regression is to select the optimal hyper-parameter λ based on Cross Validation, so the prior distribution is estimated from the data. This approach stands in contrast to standard Bayesian methods, in which the prior distribution represents the a priori information and should be fixed before observing the sample. The Bayesian procedures in which the prior distribution is estimated from the data are called *Empirical Bayes* methods.

Other Bayesian estimators for GLM

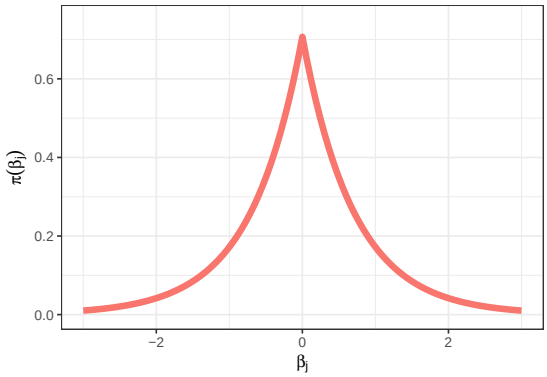
The Bayesian framework provides more flexibility in prior information than what standard Ridge and LASSO regression offer.

Even staying with the Normal prior distributions, it is possible to give to the different coefficients β_j different prior variances σ_j^2, that would represent how much we trust the estimated coefficient to be different from 0. For example it is possible to use a non informative prior for most of the coefficients and to introduce an informative prior only to the more delicate ones. That corresponds to adding a penalization only to those coefficients. As we have already discussed in section 2.1.3, this technique can be useful in the case of highly unbalanced explanatory variables, such as the variable "number of claims experienced in the previous year".

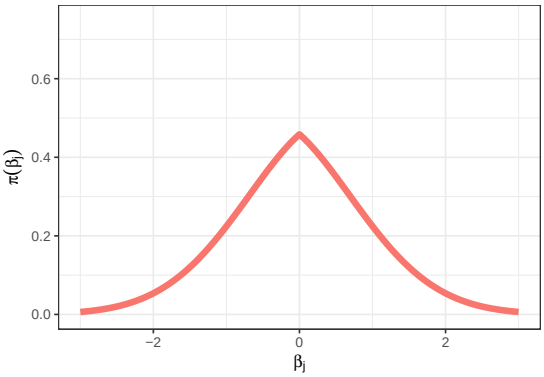
It is also possible to assign to the coefficients prior distributions not centered on 0. If we assign to the coefficient β_j a prior $\beta_j \sim \mathcal{N}(\beta_{j0}, \sigma_j^2)$, this results in a shrinkage towards β_{j0} instead than towards 0. In the Bayesian reasoning, this can be useful if we want to add to the model information taken from outside the sample. For instance,



(a) Normal distribution



(b) Laplace distribution



(c) Elastic Net prior distribution with $\alpha = \frac{1}{2}$

Figure 2.15: Normal density function, Laplace density function and Elastic Net prior density function with unitary variance.

if we want to fit a model for a small portfolio of insurance policies, we can introduce prior information from estimates from other portfolios or from market data. Given the prior distribution, the exposure of the portfolio used for fitting determines how much information we have on that portfolio and so how much our posterior estimates will be close to the Maximum Likelihood estimates.

The Bayesian framework allows us also to use prior distributions different from the Normal and the Laplace. For example, if we want a coefficient β_j to be positive, we can assign to it a prior distribution $\pi(\beta_j)$ with all the mass on positive values and density equal 0 to all the negative determinations. This will force the Maximum a Posteriori estimate to be non negative.

$$\pi(\beta_j) = \begin{cases} \frac{\sqrt{2}}{\sqrt{\pi}\sigma} e^{-\frac{1}{2\sigma^2}\beta_j^2} & \text{if } \beta_j \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

A useful application of prior distributions with all the mass on positive value is the following. Suppose we have a discrete ordinal variable with determinations encoded in the dummy variables $x_1, x_2, \ldots, x_J$. In the common GLM parametrization, each dummy variable corresponds to a coefficient $\beta_1, \beta_2, \ldots, \beta_J$. We can consider the difference between each and the previous one: $\gamma_j = \beta_j - \beta_{j-1},\ j \in \{2, 3, \ldots, J\}$. The new parametrization depends on the coefficients: $\beta_1, \gamma_2, \ldots, \gamma_J$. If we assign a prior distribution centered on 0 for the coefficients $\gamma_2, \ldots, \gamma_J$, this will produce a shrinkage of each of the coefficients $\beta_2, \ldots, \beta_J$ towards the previous one, forcing the variable effect to have small steps from one modality to the next one. Moreover, if we assign to $\gamma_2, \ldots, \gamma_J$ prior distributions with all the mass on positive determinations, the estimate will produce a monotonically non decreasing effect as:

$$\gamma_2, \ldots, \gamma_J \geq 0 \ \longrightarrow\ \beta_1 \leq \beta_2 \leq \cdots \leq \beta_J$$

In the same way, if we assign to $\gamma_2, \ldots, \gamma_J$ prior distributions with all the mass on negative determinations, the result will be a monotonically non increasing effect.

Some considerations on Bayesian estimators for GLM

Bayesian estimators offer a flexible tool for modeling effects in GLM. This tool is particularly useful in actuarial modeling because often the actuary needs to introduce external information to tune the model coefficients.

In the actuarial practice this is often done introducing offsets. For example, let's consider a delicate coefficient β_j estimated for a class with low exposure. This can happen for example with the variable "number of claims experienced in the previous year" with the class "more than one claims". If we want to insert it in the model with an effect weakened compared to the maximum likelihood one $\hat{\beta}_j^{ML}$, we can consider a specific value $\hat{\beta}_j^{\text{offset}}$, with $|\hat{\beta}_j^{\text{offset}}| \leq |\hat{\beta}_j^{ML}|$, in the model and introduce it as an offset.

With Bayesian estimators, in the practice, we can do something really similar to what we do with offsets. Indeed, we can choose a prior distribution, such that our estimate is exactly equal to the offset $\hat{\beta}_j^{MAP} = \hat{\beta}_j^{\text{offset}}$. For example, in the case of the delicate coefficient β_j, we can just apply a Normal prior $\pi(\beta_j)$ centered in 0 and tune the variance σ_j^2 in order to obtain exactly $\hat{\beta}_j^{MAP} = \hat{\beta}_j^{\text{offset}}$.

This practice look really similar to introducing an offset, but it has two important benefits.

First of all, even if we are forcing the coefficient to be exactly $\hat{\beta}_j^{MAP} = \hat{\beta}_j^{\text{offset}}$, we have an indicator on how strong our forcing is. Indeed, σ_j^2 can be seen as the strength of the prior distribution and it can be compared with the variances of the prior distributions of the other coefficients.

Secondly, in many cases, after setting an offset, we want to modify the model and eventually change other coefficients. After doing this, if we are working with offsets, we have to review each of the coefficients imposed in offset, since a change in

other coefficients can affect them. With Bayesian estimators we can just keep the prior distributions previously set and refit the model. The new coefficients will be automatically reevaluated keeping the strength of the prior distributions previously assigned.

We end our discussion on Bayesian estimators for GLM by mentioning that also GAM models, discussed in section 2.1.2, can be estimated with Bayesian estimators. This is done by applying a prior distribution to the coefficients of the spline functions $f_l(\cdot)$. For more details on that, we refer to [2].

2.2 Considerations on models

In this section we are going to discuss how the models we have seen in the previous sections satisfy the non-life insurance procing needs. In subsection 2.2.1 we will also give some hints on *Machine Learning Algorithms*, that will be useful to compare them with the models described in this thesis.

2.2.1 Hints on Machine Learning Algorithms

With the term *Machine Learning Algorithms* we refer to a set of techniques used for predictive models in Machine Learning Practice, such as GBM, Random Forest (RF) and Neural Network (NN). These are highly flexible general purpose techniques that can be used both for regression (predictive models with a quantitative response variable Y) and classification (predictive models with a qualitative response variable Y). In this section we are going to only provide some hints on them. For a more in-depth exposure of these models with applications to actuarial problems we refer to [3].

The assumptions adopted in the models behind these algorithms are minimal. Usually it is just assumed a distribution for the response variable Y, while for the regression function we just consider:

$$E(Y_i) = \boldsymbol{f}(x_{i1}, \ldots, x_{ip})$$

without any constraints of the shape of $\boldsymbol{f}(\cdot, \ldots, \cdot)$. That means that with these models we take into account all the possible interactions between variables without any restriction.

Given the distribution of Y, we are able to define a loss function, that for example can be the deviance $D(\boldsymbol{f}, \boldsymbol{y})$. The aim of the fitting algorithm is to find a good approximation $\hat{\boldsymbol{f}}$ for $\boldsymbol{f}$.

For highly complex models, there is a huge risk of overfitting the training set. These Machine Learning Techniques offer sophisticated algorithms that provides efficient way to fit $\hat{\boldsymbol{f}}$ preventing overfitting.

The focus of these techniques is usually much more on the fitting algorithm rather than on the underlying model. For these reason they are often referred as *Machine Learning Algorithms* rather than *Machine Learning Models*.

These algorithms provide a high level of automation and do not require big manual interventions by the person who is running the algorithm. In these algorithms neither the shape of the regression function nor the interactions between the explanatory variables have to to be specified.

The results of the fitting are complex regression functions $\hat{\boldsymbol{f}}$. The convenience of this complexity combined with a strong automation is that these algorithms are able to automatically spot complex effects that wouldn't be easily discovered by manual fitting.

On the other hand, this complexity reduces the interpretability of the result. For this reason, these algorithms are often referred as *Black Box*.

We underline that, from a theoretical point of view, GLM and all their advancements seen in this thesis are actually *Machine Learning Techniques*, since there is not a theoretical distinction between *Statistical Models* and *Machine Learning Models*. Anyway, in the common speech, the term *Machine Learning* is used for algorithms such as GBM, Random Forest and Neural Network, that, compared to classical statistical models, offer more complex estimations and a higher level of automation. In the following, we will use the term *Machine Learning Algorithms* to refer to algorithms such as GBM, Random Forest and Neural Network.

2.2.2 Model comparison

In this thesis we have discussed some GLM advancements: GAMs (section 2.1.2), Shrinkage Estimators (section 2.1.3) and Bayesian Estimators (section 2.1.4). All these developments offer better predictions and more automation compared to classic GLM. For example, in GAM fitting it is not needed to specify the shape of the effect for the quantitative variables and in Elastic Net an automatic variable selection is performed. The improvement in performance both in term of predictivity and automation increases as the number of explanatory variable increases, so these techniques become significantly better than classic GLM when p is big. As we have seen, these techniques are not mutual exclusive. It is possible to fit a GAM with penalized splines for quantitative explanatory variables and Elastic Net penalization for qualitative explanatory variables. The Bayesian estimators, that can be seen as a generalization of Elastic Net, can be applied to both GLM and GAM and offer even more useful tools for fitting.

Compared to these GLM Advancements, Machine Learning Algorithms are still much more flexible and automatic. Indeed, in GAM and Elastic Net there still is a certain degree of manual supervision. For example, we still have to specify the interactions we want to consider in the model. Actually, in Elastic Net we can consider a large number of interactions and let the model automatically select them, but it is still preferred to check what the model is doing and eventually manually choose whether to include or not the selected interactions.

Anyway, Machine Learning Algorithms are much less *interpretable* and *controllable* compared to GLM-based models. In GLM-based models we can easily look at the marginal effects of the explanatory variables and we have a full control of the interactions. In Machine Learning Algorithms, we can't easily look at how the changing of an explanatory variable affects the estimate on the response because the effect of that changing can highly depend on the values of the other explanatory variables in a way that we can't manually control.

However, the benefits of GLM-based models go beyond the interpretability. In GLM-based models, the person who is running the fitting, have a *full control on the coefficients*.

Indeed, he can easily choose to add or not an effect regardless the variable selection procedure includes it or not. Furthermore, the choice consists not only in adding or not a variable or an interaction, but, in GLM-based models, it is even possible to manually assign to an explanatory variable a marginal effect with a specific shape by introducing an offset or, with Bayesian estimators, by tuning the prior distribution.

2.2.3 The actuary importance

The high level of discretion in GLM-based models fitting gives a lot of importance to the person who is fitting the model. In actuarial models, the person that conducts the model fitting is the *actuary*. In section 1.5 we have seen who the actuary is. In this section we will describe some of the cases in which a manual intervention in the model fitting is needed and how the actuary can perform it.

We will distinguish between *technical needs*, that are related to building a good predictive model, and *commercial needs*, that are related to price optimization.

Technical needs

The *technical needs* come from the fact that we observe past data, but our models will be used for pricing policies that will be sold in the future. So, our models have to combine observation from the past and assumptions on the future.

By building a technical price, we must consider that our portfolio is not representative of the whole country portfolio and it isn't even representative of our future portfolio, that is influenced by the future underwriting policy of the company and also by the pricing policy.

There are some cases in which estimates from past data could lead to a *bias* in the future predictions. For example, it is possible that in the past the underwriting policy for certain clusters, such as customers from specific regions, has been particularly strict, so on those clusters, on historical data we see excellent technical results. Anyway, it is possible for these past observations to not be representative of the future policies we could acquire from those clusters, so, proposing a price too low for customers from those clusters could result in big technical losses. On the other hand, it is possible that in the past the company suffered big frauds from customers of specific clusters, but, as the fraud detection initiatives improved, in the future it is not expected to acquire fraudolent customers from those clusters. In these cases, the coefficients fitted from past data will be too high for future policies from those clusters.

It is also possible that in the future the company is going to sell on new sales channels where there is no historical data. It is possible that the policies from the new channels will have different characteristics, such as an higher or a lower propensity to commit frauds, so the actuary should take it into account with a proper pricing correction.

Moreover, it is possible that in the future the company will change the way some guarantees are sold. For instance, if in the past the option of payment division into installments was not encouraged, only the bad customers with high premiums would have chosen to divide their payment into installments. But, if in the future that option will be encouraged, a wider range of customers will select it and the determination of the coefficient of the variable "number of installments" will change.

There are other cases in which the past portfolio on certain clusters is too small, so the estimates with past data are affected by *high variance* and they are not reliable. This can happen for example if in a specific region the company never pushed and has a small market share.

In all the cases mentioned, the actuary has to use his expertise to properly pricing policies. In GLM-based models, looking to the fitted marginal effects of the critical variables and manually changing them is quite straightforward, while it would be much more difficult for complex machine learning models.

Commercial needs

As we have seen in section 1.4, the tariff and the offer price are the result of a process of *price optimization* that takes into account technical pricing, client expectation and business strategy. Usually the definition of the tariff and the offer price starts from the technical price and from it some of the coefficients are tuned to satisfy commercial needs.

The tariff and the offer price must also respect specific legal constraints. One example of legal constraint in MTPL coefficients is related to the bonus-malus class. By law, the coefficients of the bonus-malus class must be monotonically increasing. That means that, even if in the technical price the fitted coefficients are not increasing in all clusters, in tariff and offer price they must be tuned to achive monotonicity.

Some commercial constraints that are logical for tariff and offer price are the ones on coefficients related to coverage options such as the insurance ceiling and the deductible. It is clear that, if a client asks for a higher ceiling, the price offered to him must be higher. However, it is possible that on observed data on average the policies with higher ceiling experience less claims and have a lower total cost of claims. This can be explained with the fact that policyholders that select the higher ceiling are usually the more careful, so they commit less claims. The same way, if the client asks for a higher deductible, the price offered to him must be lower, even if it is possible that policies with higher deductible have an higher technical price that policies with a lower deductible.

We must also consider commercial constraints based merely on customer expectation. For example, customers usually expect that insuring a more powerful vehicles should be more expensive than a less powerful one. From observed data it is possible that policies with more powerful vehicles have on average a lower total cost of claims. Moreover, the power of a vehicle is positively correlated with its commercial price and customers that buy expensive vehicles are usually the less sensitive to price. With

these consideration, from a commercial point of view, it is preferred to have an offer with higher prices for more powerful vehicles.

Another set of variables that have to respect some commercial constraints are the variables related to claims experienced in the previous years. The effect of those variables should follow a specific order. Indeed, a claim experienced last year must penalize the premium more than a claim experienced two years ago, that must penalize more than a claim experienced three years ago and so on. On top of them it is also need to take into account the period of coverage. For instance, let's consider a customer that experienced a claim last year and previously has been covered for 5 years with no claims, and another customer that experienced a claim last year but has never been covered before. If all the other characteristic for the two customers are the same, the first one should receive a lower price than the second one. These constraints, in addition to being guided by commercial logic, also make sense from a technical point of view. Checking whether these constraints are always respected is feasible with GLM-based models, but it is quite difficult for complex machine learning models.

Finally, we mention that there are some practical cases in which it is important to understand why the offer price results in a certain value. For instance, if a customer makes a variation on the policy, for example by changing the insured vehicle, and the price changes significantly, it is important to understand which variables caused the change. This is important also to understand if the change is intentional and we really think that with the other vehicle the expected total cost of claims is different or for example the result comes from an information technology issue, such as that a variable is not properly passed to the calculation engine.

2.3 Implementation

2.3.1 Practical data problem and solutions

In building actuarial models we also face practical challenges due to the size of the data we are working with. In modern actuarial modeling it is common to work with datasets with millions of rows and hundreds of columns, that can weight several Gigabytes. With legacy Information Technology (IT) implementations, dealing with these kinds of datasets can be quite painful. The issues appear not only in fitting models, but in the whole Extract Transform Load (ETL) pipeline. Working with big amount of data is struggling on two main aspects:

1. Memory;
2. Computation time.

The first problem consists in the fact that in a system the amount of primary memory, i.e. the Random Access Memory (RAM), is limited and can't be arbitrarily increased. When the datasets we use are larger than the RAM in our local machine, it is not possible to work with them in memory locally. A possible solution is to keep the data on the secondary memory, i.e. the Hard Disk Drive (HDD) or Solid State Drive (SSD), and read it sequentially only when it is needed, as it is done in some analytics software like SAS. This solution permits to work with larger datasets, but implicates a significant decrease in computation speed, since reading and writing on secondary memory is much slower than doing it on primary memory.

Considering the computation time, even if our data fits into the primary memory, if we want to run a very complex algorithm with our processor, it could be infeasible in a reasonable amount of time. Indeed, the speed on a single Central Processing Unit (CPU) is limited by physical constraints.

In modern computer science applications, the solution to these problems comes from the concept of parallelism. Parallel computing is a type of computation where many calculations are carried out simultaneously. This can be achieved with two paradigms:

1. *Task parallelism*;
2. *Data parallelism*.

Task parallelism is what is done with multi-core processors. A multi-core processor is a processor that includes multiple processing units, called "cores", on the same chip. The advantage of having a machine with multiple cores is that it is possible to decompose our algorithm into smaller tasks that can be run simultaneously. For instance, when we are conducting hyper-parameter tuning in a model, we are fitting the model many times with different values of the parameters. These fittings are independent tasks that can be conducted simultaneously by the different cores of our machine. Modern computers are equipped with at least two cores and many statistical packages deploy task parallelization to reduce computation time.

Anyway, task parallelization does not tackle the problem of limited available memory. If our dataset is bigger than our primary memory, we can consider *data parallelization*. Data parallelization consists in distributing the data across different machines, which operate on the data in parallel. By distributing the data, each machine has only a portion of it and, if there are enough machines, even with a large dataset, it is possible to split it into pieces that fit into the primary memory of the single machines. The set of connected machines that work together in a distributed system is called *cluster*. Cluster computing offers a reliable and flexible way to deal with big datasets in analytics. The greater advantage of cluster computing is that it is easily scalable, that is, if we need more memory or more computing power, we can just increase the number of machines in the cluster and increase the performance of the system.

Moving from a centralized system to a distributed system requires an important level of sophistication from an algorithm point of view. Luckily, there are many frameworks such as Apache Hadoop and Spark, that offer high-level interfaces that can be easily deployed by the final user. The high-level of abstraction of these interfaces permits to the final user to conduct the analysis in a distributed system without worrying on the low-level implementation details.

An even higher level of abstraction is achieved with *cloud computing*. Cloud computing consists in the on-demand availability of computer system resources, especially data storage and computing power, without direct active management by the user. The idea behind cloud computing is to demand the whole IT management to a third company and paying to it a fee based on the use. This paradigm is also called Hardware as a Service (HaaS), as the client doesn't own the hardware and just accesses the storage and the computing power when he needs it. With this approach, scalability is even easier, because, if the user needs more resources, he can just ask for them and the service provider will quickly satisfy his request. With own hardware, increasing the hardware resources would require buying new hardware and installing it, that could be a big investment both in term of economic cost and time spent.

We mention that, even if the solutions mentioned in this chapter are suitable for *big data*, in usual actuarial pricing, big data are not used. The characteristics usually associated with big data are *Volume*, *Variety* and *Velocity*. In common actuarial practice variety and velocity are not contemplated, as actuarial datasets are usually just relational tables that, at the moment of modeling, are previously built. The situation changes by introducing real time data such as telematics car driving data. For instance, if we want to model the real-time risk of committing a claim based on data from speedometers, accelerometers, gyroscopes and GPS that capture how and where the vehicle is driven, that would be a big data problem.

2.3.2 Actuarial pricing specific needs

In actuarial practice, as the actuary needs to visualize, interpret and control the variables effects, it is also important how the interface works. Often visual point and click applications are preferred to coding interfaces, such as R and Python. While coding interfaces are often used for ETL processes for their flexibility, in the modeling process point and click applications are considered more user friendly, since they require less effort from user to become proficient with and they often offer out of the box tools to easily visualize results.

Among the point and click applications, in the market there are some solutions for actuarial pricing that are specifically thought for building GLMs and authomatically produces the plots seen in section 2.1.1. These applications are particularly optimized for GLM, but they are not as flexible as machine learning libraries in R and Python and the advancement implemented are limited.

2.3.3 Solution adopted in this project

In the application described in section 3 we used H2O [4]. H2O is an open-source machine learning platform that provides many of the most widely used statistics and machine learning models, such as GLM, GAM, GBM, Random Forest and Neural Network. H2O engine is based on Java, but it can be used both in R and Python with the packages deployed for those languages. The packages work as interfaces that talks with the Java engine, but still keeping an overall syntax consistency with the programming language we are using. That means that we can use H2O both on R package and Python package and the engine that is running under the hood is exactly the same.

H2O framework offers also H2O Flow, that is a point and click interface that allows users to run H2O machine learning algorithms without writing code. This is a user friendly interface for modeling, but it has not all the functionalities for visualization, interpretation and control of variables effects that are present in actuarial pricing specific applications mentioned in section 2.3.2.

As well as having several interfaces, H2O can run both locally on a single machine, with multi-core processing, and on a Spark Cluster. The high-level of abstraction of the interfaces permit to develop the algorithms both locally and on a cluster with basically the same code.

In the application described in section 3 we used H2O locally on R.

CHAPTER 3

Practical application

In this final chapter we are going to describe the practical application on an actuarial dataset of the models described in section 2. After describing the problem and how the models have been implemented, we will assess their performance and comment the results.

3.1 Data description

3.1.1 Dataset

The dataset used is an actuarial dataset from an Italian MTPL portfolio. In table 3.1 the total exposure of the dataset is reported. In the dataset there are 283 829 rows for a total of 134 804.7 years-at-risk. In the dataset, every row is a couple (policy, accounting year), so, if a policy spans in two years, we will see two rows corresponding to that policy. On average, every row has exposure 0.475 years-at-risk. This number is slightly lower than 0.5 because some of the policies get suspended and span in more than two years.

Table 3.1: Total dataset exposure. The exposure is measured in year-at-risk (Y.a.R)

Observations	Exposure (Y.a.R.)	Average Exposure per Observation
283 829	134 804.7	0.475

The dataset contains policies from the period 2014-2019 coming from a specific

province. The fact that we filtered the policies from only one province reduces a lot the size of the dataset and, since the models has been run locally on a single computer, this significantly reduces the time spent to run them. From a modeling point of view, the reduction to only one province reduces the importance of the geography in the models. Indeed, between the different regions of Italy there are a lot of socio-economical differences that are reflected to the claims experience. In this thesis the problem of using geographical data as explanatory variables has not been addressed.

The dataset has been split into a training set with 80% of the observations and a test set with the remaining 20% of the observations. The training set has been used to fit the models, while the test set has been be used to assess them by comparing the predictions of the models with the observed data. The splitting has been made on the base of the policyholder id. That means that policies from the same policyholder end in the same set. This has been done because policies from the same policyholder are correlated, so splitting the observations by policyholder prevents data leakage from the test set to the training set. Table 3.2 shows the exposure and the number of policyholders in the training set and the test set. As we can see, on average, each policyholder has 3.95 years-at-risk.

Table 3.2: Exposure and number of policyholders in training and test set. The exposure is measured in year-at-risk (Y.a.R)

Set	Observations	Exposure (Y.a.R.)	Policyholders	Exposure per Policyholder
Train	227 226	107 998.4	27 346	3.95
Test	56 603	26 806.3	6 824	3.93
Tot	283 829	134 804.7	34 170	3.95

3.1.2 Response variable

The response variable for the models is the number of claims caused by the policyholder for that policy in that accounting year. In this project we compared the different techniques in predicting the claims frequency. We didn't take into account the claim severity and we didn't split the claims by typology (see section 1.3.4).

Table 3.3 shows the observed claims frequency in the training set and in the test set. As we can see, the claims frequency in the training set is slightly higher than the claims frequency in the test set. The difference between the two sets is given by chance, but it is also encouraged by the fact that the policies has been split by policyholder id and not by policy. The split by policyholder id implicates that, if a policyholder has many policies, all of them fall into the same set, so the hypothesis of independence for the observations within the sets is not respected. Anyway this issue is restrained

and in this analysis has been neglected, as it is usually done in actuarial practice. We also mention that theoretically the difference in claims frequency in the two sets should be mostly explained by the explanatory variables we have, so the models fitted with the training set should be able to catch the characteristics of the policies of the test set that reduce their expected claims frequency.

Table 3.3: Claims Frequency in training and test set.

Set	Observations	Exposure (Y.a.R.)	Claims N.	Claims Freq.
Train	227 226	107 998.4	4 823	0.045
Test	56 603	26 806.3	1 131	0.042
Tot	283 829	134 804.7	5 954	0.044

As we mentioned in section 1.3.5, the claims settlement is a long process and the response variable we use for fitting the models depends on the moment in which the claims are observed. Indeed it is possible that part of the claims we are considering as response variables will be closed as null-claims and it is possible that there are claims occurred in the period of exposure considered that are not yet reported (IBNyR) at the moment in which the claims have been observed. This phenomenon is more relevant if the period of exposure is close to the moment of observation of the claims.

The claims reported in the dataset are all observed at the date 30/06/2020. In figure 3.1 the claims frequency for each year in the training set and in the test set is reported. The same data is reported in table 3.4. As we can see the years 2017, 2018 and 2019 present a claims frequency lower than the years 2014, 2015 and 2018, so it looks like there is a structure in the claims frequency over the years. Unfortunately, just by looking to this data we can't say whether this is due to the different settlement of the claims over the years, to a different mix of policies or just to a overall claims trend in the whole market. Anyway, looking to the confidence intervals we realize that this is not a big issue as the effect of the year is not too heavy.

Comparing the claims frequency of the training set and the test set, we see that the claims frequency in the test set is consistently lower than the claims frequency in the training set. This phenomenon can be explained with the fact that we divided the policies based on policyholder id, so the policies within each set are correlated.

Exploration on the effect of splitting data by policyholder

To explore more in depth the effect of splitting data into training and test set by observation or by policyholder, we conducted a simulation analysis. We took the whole dataset and we simulated for each observation a random flag that represents whether the observation belongs to the training set or to the test set with 80% probability of belonging

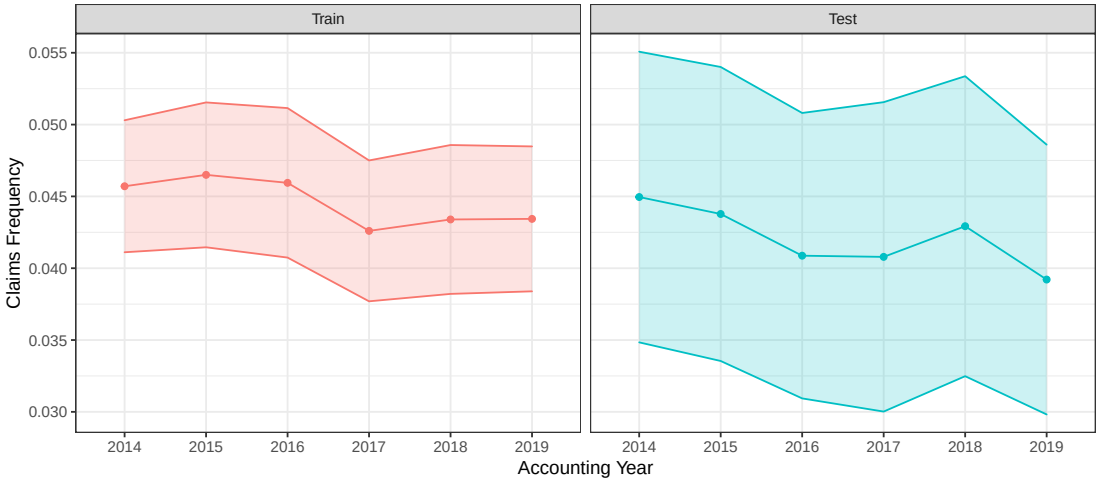


Figure 3.1: Claims frequency in training and test set in the different years. The semi-transparent ribbons represent 95% confidence intervals for the claims frequency in each year.

Table 3.4: Claims frequency in training and test set in the different years.

Set	2014	2015	2016	2017	2018	2019
Train	0.046	0.046	0.046	0.043	0.043	0.043
Test	0.045	0.044	0.041	0.041	0.043	0.039
Tot	0.046	0.046	0.045	0.042	0.043	0.043

to the training set and 20% probability of belonging to the test set. We repeated the simulation 10k times with a random variable based on observation and 10k times with a random variable based on policyholder. In each simulation, we fitted a GLM with only that flag as explanatory variable and the number of claims as response variable. Then, for each simulation, we looked at the p-value of the test over the significance of that flag. This p-value can be seen as a measure for checking whether the training and the test sets have a significative difference in term of claim frequency.

The distribution of the p-values for the two kinds of random variables are reported in figure 3.2. The frequency for the first 4 buckets are also reported in table 3.5. In theory, if the variable was totally random, the distribution of the p-values should be uniform. As we can see, the distribution of the p-values for the random by policyholder is not uniform and the lower p-values are more likely. Indeed, in our simulation, the observed frequency of the p-values lower than 0.05 with the random by policyholder is 0.0782, that is quite higher than 0.05. Anyway, we accept this potential difference in the claims frequency between the training set and the test set and we keep the splitting made with the random by policyholder.

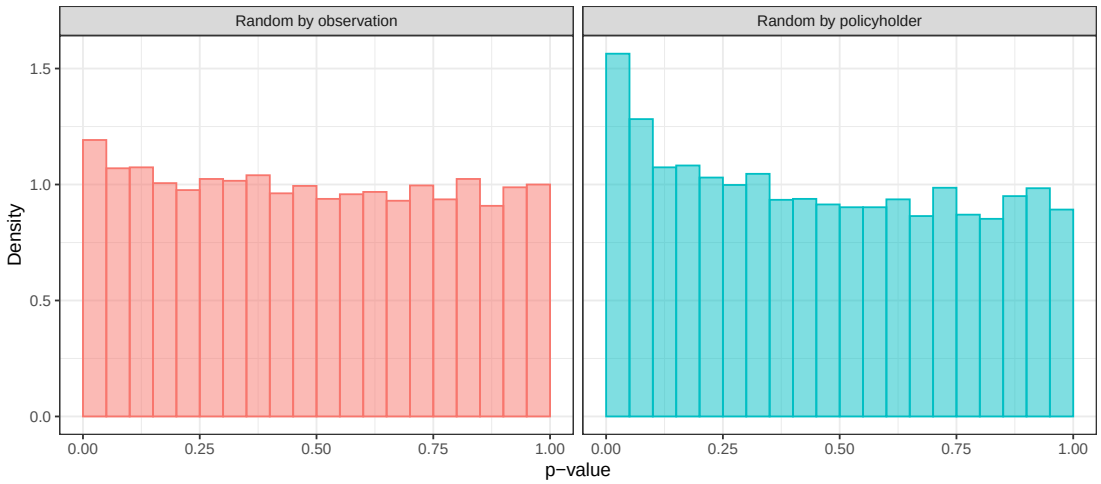


Figure 3.2: Distribution of the p-values from the simulation of the random variables. With the random by policyholder, the lower p-values are more likely.

Table 3.5: Distribution of the p-values from the simulation of the random variables. With a random variable, these frequencies should be around 0.05. With the random by policyholder, the frequencies for low p-values are higher than that.

Random type	**[0, 0.05)**	**[0.05, 0.1)**	**[0.1, 0.15)**	**[0.15, 0.2)**
Random by observation	0.0596	0.0535	0.0537	0.0503
Random by policyholder	0.0782	0.0641	0.0537	0.0541

3.1.3 Explanatory variables

In the dataset we have a total of 52 explanatory variables that are considered meaningful for predicting the claims frequency. In table 3.6 we can see the number of variables split by category. The categories are the ones described in section 1.3.1.

In the modeling process we will assess whether these variables are significative or not and only some of them will be used to predict the response variable.

Table 3.6: Number of explanatory variables per category.

Description	Number of variables per category
Information on the insured vehicle	12
General information of the policyholder	14
Insurance specific information of the policyholder	9
Policy options	11
Customer information on the policyholder	2
Telematic data	4
Total	**52**

3.2 Models assessment

The splitting of the dataset into training set and test set allow us to use the training set to fit the model and the test set to assess them by comparing the predictions of the models with the observed data.

The metric adopted to assess the performance of each model is the Poisson deviance (see section 2.1.1 for more details):

$$D(\hat{\boldsymbol{\beta}}, \boldsymbol{y}) = 2 \sum_{i=1}^{n} \left\{ y_i \log \left(\frac{y_i}{\hat{\mu}_i} \right) - (y_i - \hat{\mu}_i) \right\}$$

The lower the deviance on the test set is, the better the model is.

3.3 Models description

3.3.1 List of models

In table 3.7 the models developed are listed. We started from a GLM that takes into account all the variables available (*GLM Tot*). Then, with the same variables, we considered an Elastic Net (*Elastic Net Tot*) and a Ridge regression (*Ridge Tot*). After this, we computed a feature selection with a stepwise approach based on AIC obtaining a GLM with just a subset of the available variables (*GLM AIC*). With the same subset of variables we also fitted an Elastic Net (*Elastic Net AIC*) and a GAM (*GAM AIC*). Finally, to have a comparison with a widely used general purpose machine learning model, we fitted a GBM with all the variables available (*GBM Tot*). More details on the models listed in table 3.7 are described more in depth in section 3.3.4.

Table 3.7: List of models developed.

Id	Model
Mod1	GLM Tot
Mod2	Elastic Net Tot
Mod3	Ridge Tot
Mod4	GLM AIC
Mod5	Elastic Net AIC
Mod6	GAM AIC
Mod7	GBM Tot

3.3.2 Implementation details

All the models have been fitted with H2O, except for the GAM. On H2O the following functions are available:

- `h2o.glm()` that allows to build GLMs with Elastic Net penalization;
- `h2o.gam()` that allows to build GAMs with both GAM penalization to splines and Elastic Net penalization to other variables;
- `h2o.gbm()` that allows to build GBMs.

Anyway, as described in H2O documentation, "GAM models are currently experimental". We had some trials with GAMs on H2O, but there are still many bugs and many functions for the hyperparameter tuning are not available yet. For this reason, for the GAM model we used the function `gam()` from the package `mgcv`.

The big advantage of H2O is that it provides embedded multi-threading implementation and provides built-in functions for hyper-parameter tuning. In the practice, that translates to a high-performance solution that is substantially faster and is much more memory-efficient than what the other R packages offer.

Due to the lower efficiency and the lack of hyper-parameter tuning functions, we used the function from `mgcv` just as a quick trial, without performing a deep hyper-parameter tuning.

Al the pre-processing and the model fitting has been run locally on a single PC with 16GB of DDR4 RAM and a CPU Intel Core i7-8750H, that is a 2.20GHz hexa-core CPU with hyper-threading. Within these resources, 12GB of the 16GB available RAM and 11 of the 12 available threads have been dedicated to the H2O instance.

3.3.3 Approach adopted in the modeling

One of the strengths of the GLM-based methods is that they allows to extensively introduce external information in the process of fitting. That means that, still using the same techniques, a skilled modeler is able to build a better model than a beginner.

The aim of our analysis is to built a comparison just between the models, so the skills of the modeler would be a disturbance factor. In order to prevent this factor to influence our results, in the whole modeling process we adopted techniques as much objective as possible.

To achieve this goal in the basic GLM, for the feature selection we performed a stepwise algorithm based on AIC instead than a manual process.

The only manual element in the model fitting has been the definition of the effects of the quantitative variables. For the quantitative variables we considered split-wise polynomials effects based on the plots of the residuals. Anyway we have been quite generous in the number of terms taken into account and we let the algorithm choosing which of them to keep in the model.

For the interactions, we decided to not consider any interaction term in the GLM-based models. This choice comes from the fact that from our exploration there were not particularly important interactions between the variables and from the fact that, in GLM-based models, the choices of interactions terms is usually a manual process and the ability of the modeler influences a lot the result, introducing a disturbance factor.

We will return on some considerations of the benefits of the modeler intervention in section 3.5.

3.3.4 Models details

GLM Tot

In the first GLM we considered all the 52 explanatory variables listed in table 3.6. Considering all the levels of the qualitative variables and all the polynomial terms for the quantitative components, it results into 121 free parameters[1].

The fitting in H2O has been quite fast and it took just 2.7 seconds.

Elastic Net Tot

With the same parametrization of the first GLM, we fitted an Elastic Net. To choose the optimal parameters α and λ we computed an grid search with a cross validation. In the cross validation we tested 363 models with a total computation time of 1h 30m.

The figure 3.3 shows the Cross Validation Deviance for the different sets of hyper-parameters. As we can see, for each value of α the points draw a curve that starts high, decreases and, after an elbow, increases again. The light blue vertical line corresponds to the optimal set of hyper-parameters that are $\alpha = 0.06$ and $\lambda = 2.01e-04$.

In the plot there are some gaps between the points because, in order to reduce the execution time, the algorithm computed has been a random grid search, that means that just a random subset of grid has been tested. The time constraint of 1h 30m was an input of the algorithm.

One aspect we must underline is that, while the split between training set and test set has been made by policy id, the split into the sets of the cross validations has been made by observation. This implies that the hyper-parameter set that results from the cross validation could not be the optimal one for predicting the test set. This choice has been made because in the H2O base implementation it is not possible to specify an id for the splitting in the cross validation and we don't expect this aspect to deeply influence our results.

Within the 121 free parameters of the GLM, in the optimal Elastic Net 22 of them are shrunk to exactly 0.

Ridge Regression Tot

In figure 3.3, the red vertical line corresponds to the optimal point subjected to $\alpha = 0$. This point corresponds to the Ridge Regression solution. In the Ridge regression all the coefficients are different from 0.

[1] Within these 121 parameters we are not counting the parameters corresponding to the base levels of the qualitative variables that in the GLM usual parametrization are set to exactly 0 and we are considering the intercept.

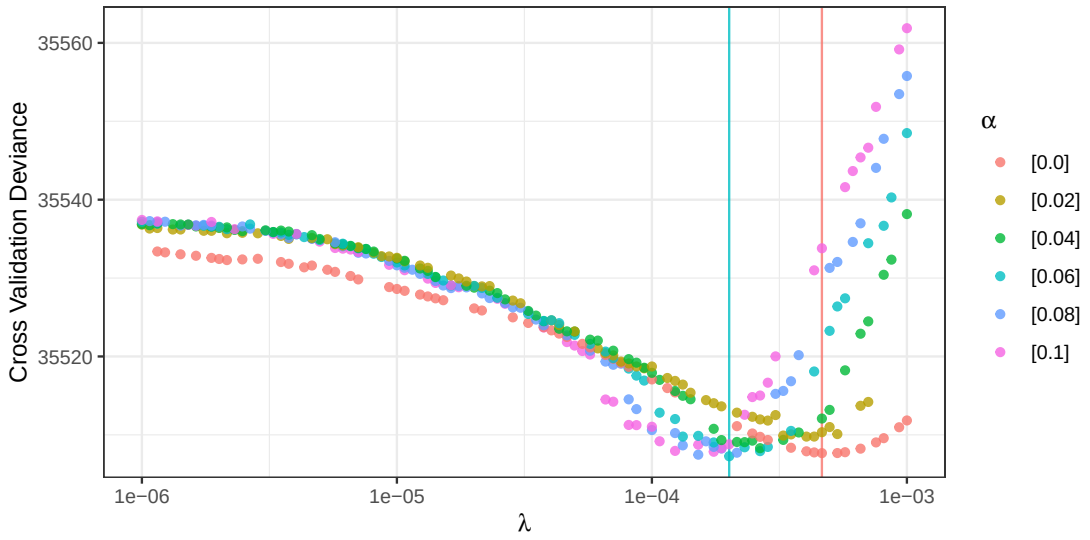


Figure 3.3: Elastic Net Tot hyper-parameter tuning. The light blue vertical line corresponds to the optimal point. The red vertical line corresponds to the optimal point subjected to $\alpha = 0$, i.e. the Ridge Regression solution.

GLM AIC

Starting from the GLM with all the explanatory variables, we applied a stepwise algorithm based on AIC to find the optimal GLM. In the model comparison we decided to move only in backward direction, since considering at each step both the forward and the backward option would have resulted in a too long process and because from some trials we found that it would have probably resulted into the exact same model.

H2O doesn't support a function for stepwise regression, so we ran the algorithm in R with the function `stepAIC()` from the package `MASS`. Running the algorithm in R is much less efficient than in H2O and it took a total time of 7h 27m.

In the final GLM obtained with the stepwise algorithm, within the 121 free parameters of the initial GLM, 64 of them have been removed from the model and only 57 lasted in the model. In table 3.8 the number of parameters equal to 0 in each model is listed. The horizontal line separate the free parameters of the initial GLM from the parameters fixed to 0 in it. From this table we can see that there are 6 parameters that are set equal to 0 in the Elastic Net Tot and not in the GLM AIC and there are 48 that are set equal to 0 in the GLM AIC and not in the Elastic Net. That means that the two feature selection criteria lead to different subsets of the explanatory variables.

Table 3.8: Parameters equal to 0 in the models.

GLM Tot	Elastic Net Tot	GLM AIC	n	
$\neq 0$	$\neq 0$	$\neq 0$	51	121
$\neq 0$	$\neq 0$	0	48	
$\neq 0$	0	$\neq 0$	6	
$\neq 0$	0	0	16	
0	$\neq 0$	0	23	38
0	0	0	15	

Elastic Net AIC

With only the variables obtained with the stepwise selection, we fitted an Elastic Net. The figure 3.4 shows the Cross Validation Deviance for the different sets of the hyper-parameters. In the cross validation we tested 600 models with a total computation time of 1h 26m.

The red vertical line in figure 3.4 corresponds to the optimal set of hyper-parameters that are $\alpha = 0$ and $\lambda = 1.63e-05$. In this case the best Elastic Net model corresponds to the Ridge Regression. That means that no coefficient has been shrunk to exactly 0. This is probably due to the fact that the stepwise algorithm already selected a set of variables that have a significative effect for predicting the response. Moreover, comparing the Cross Validation Deviance of the optimal point and the points with lower λ, we see that there isn't a substantial difference. That means that a classic GLM would perform more or less the same as the optimal Ridge Regression.

Considering all the GLM-based models we developed, we compared the coefficients estimated. The scatterplot and the correlation matrix of the coefficients are represented in figure 3.5. Here we considered also the base levels of the qualitative explanatory variables that are set to exactly 0 by the classic GLM models and we excluded the intercept, resulting into 158 coefficients.

As we can see, all the models have coefficients positively correlated. However, the most correlated are the GLM Tot with the GLM AIC and the Elastic Net Tot with the Ridge Tot.

GAM AIC

With the variables obtained with the stepwise selection, we also fitted a GAM. As already mentioned, due to the still experimental state of the GAM modeling in H2O, we fitted the GAM in R and we just took a quick trial without a deep hyper-parameters tuning. The

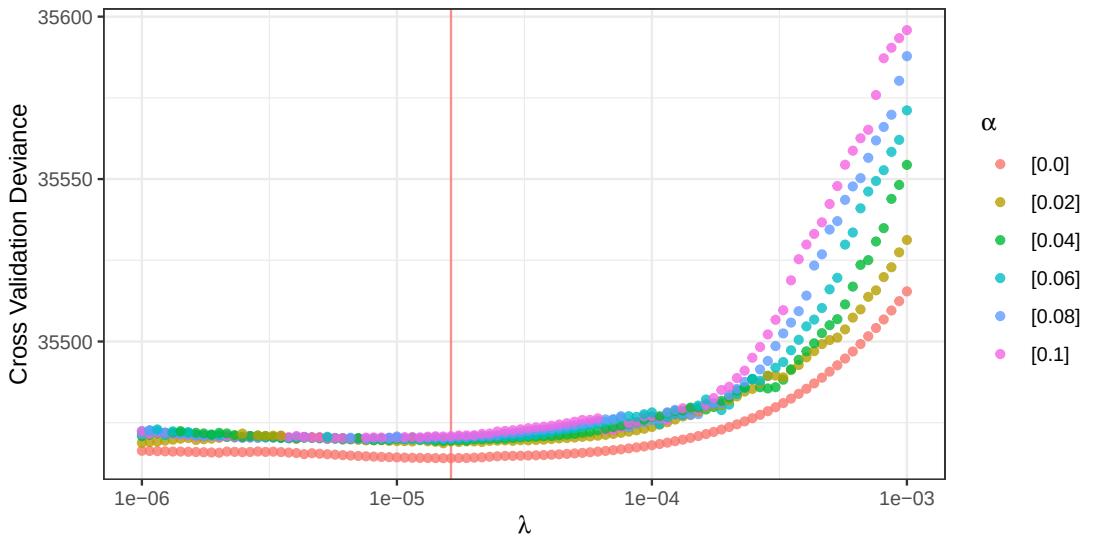


Figure 3.4: Elastic Net AIC hyper-parameter tuning.

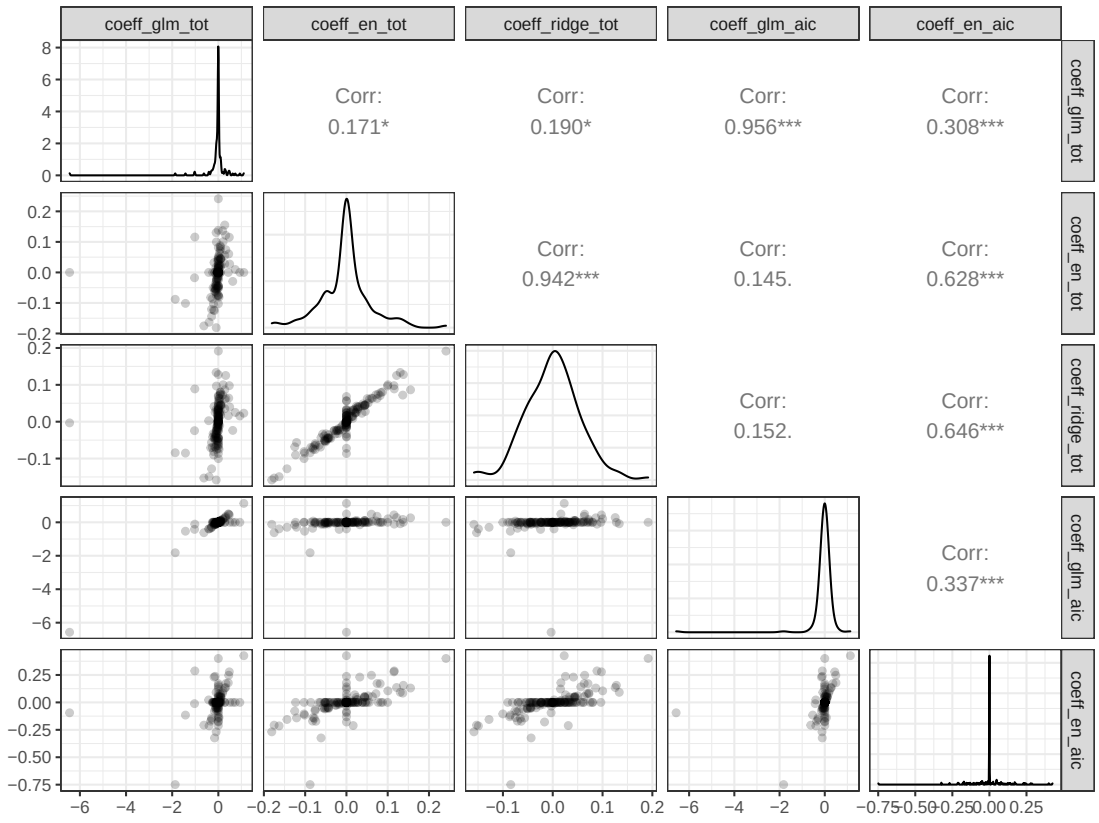


Figure 3.5: Coefficients comparison between GLM-based models.

fitting of one single GAM in R took 17m 36.7s, that means that a deep hyper-parameters tuning would have taken many hours to run.

GBM Tot

Finally, we fitted a GBM with all the 52 variables available. The GBM is one of the most widely used general purpose machine learning models. One of its befits is that it automatically performs the feature selection and automatically considers the interactions.

The GBM have some hyper-parameters that are: `ntrees`, `max_depth`, `learn_rate`, `sample_rate` and `col_sample_rate`. To understand what these parameters are we refer to the H2O documentation. For selecting the best set of variables we computed a grid search with a Cross Validation. Figure 3.6 shows the Cross Validation Deviance for the different sets of hyper-parameters. The optimal set of hyper-parameters is: `ntrees` = 408, `max_depth` = 3, `learn_rate` = 0.116, `sample_rate` = 1 and `col_sample_rate` = 1.

In total, 189 set of hyper-parameters have been tested in a total amount of time of 2h 30m.

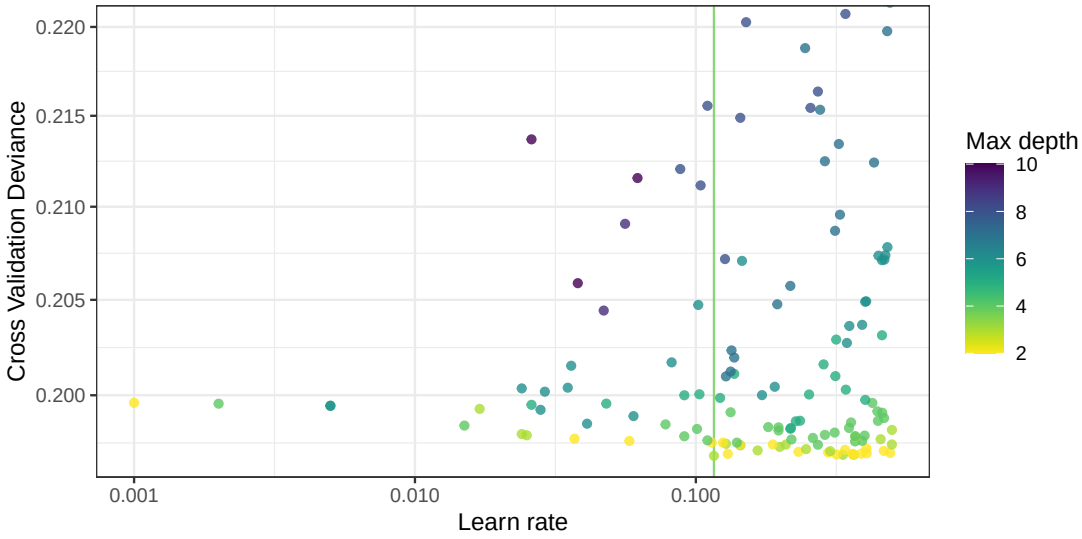


Figure 3.6: GBM Tot hyper-parameter tuning.

3.4 Results

In table 3.9 the results of the models are reported. The Test Deviance is also represented in figure 3.7. As we can see, the best model in term of Test Deviance has been the Ridge Regression with all the variables (Ridge Tot). The fact that the Ridge Regression performed the best, means that also in the variables that the Elastic Net and the stepwise based on AIC set to exactly 0 there is some useful information. As expected, the GAM performed much worse than the other models. Further experiments should be conducted to understand whether in general the GAM performs worse on this data or with a proper hyper-parameter tuning it could perform as the GLM or even better. One unexpected result is the worse performance of the GBM compared to GLM-based models. As the GBM was free to also consider the interactions, we expected it to outperform the GLM-based models. Probably in our data the interaction terms don't bring too much information to the model.

In table 3.9 also the execution time is reported. For the models based on AIC the reported time is given by the sum of the execution time of the stepwise algorithm (7h 27m) and the execution time of the further fitting. Comparing the models by the execution time, we see that running an Elastic Net on the whole dataset is much more efficient than running a stepwise algorithm based on AIC. This difference in efficiency would get even bigger by increasing the number of variables in the model, as the complexity of the stepwise algorithm grows exponentially with the number of explanatory variables.

Conducting a cost-benefit analysis with these results, we can say that, with this implementation setting, the increasing of execution time of the GLM advancements is huge and, even if it is just machine time and not human time dedicated to a manual work, it could not be justified by the small decrease of deviance. Anyway, adopting solutions with higher performance, such us the cluster computing, this trade off could considerably move towards the more sophisticated techniques.

Table 3.9: Models results. In the columns we reported the deviance computed in the test set, the execution time and, for the GLM-based models, the hyper-parameters.

Id	Model	Test Deviance	Time	α	λ
Mod1	GLM Tot	8 458.147	2.7s	0	0
Mod2	Elastic Net Tot	8 458.024	1h 30m	0.06	2.01e-04
Mod3	Ridge Tot	**8 457.465**	1h 30m	0	4.64e-04
Mod4	GLM AIC	8 458.023	7h 27m	0	0
Mod5	Elastic Net AIC	8 458.236	8h 54m	0	1.63e-05
Mod6	GAM AIC	9 728.570	7h 45m	0	0
Mod7	GBM Tot	8 504.178	2h 30m		

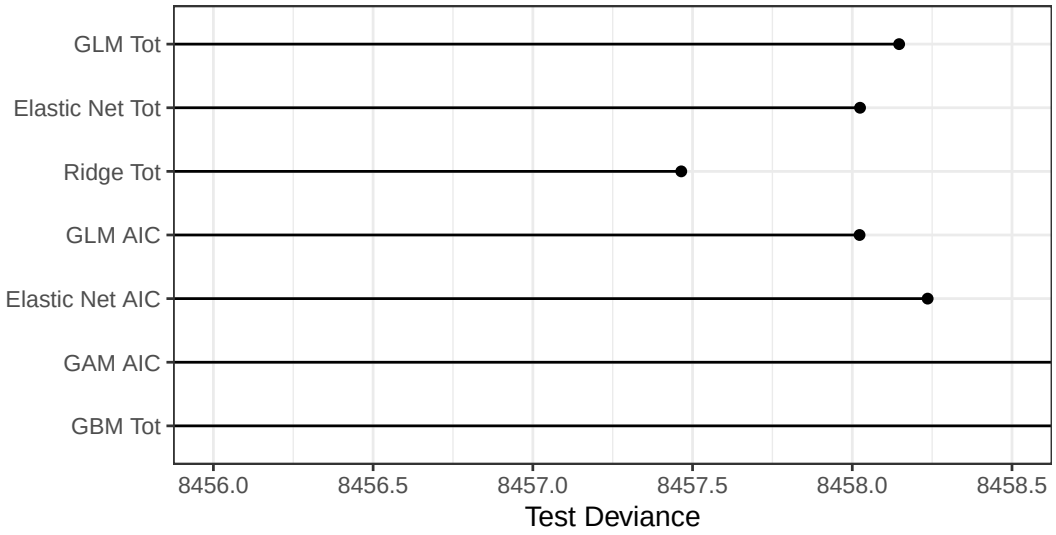


Figure 3.7: Deviance computed in the test set for the models considered.

3.5 Conclusions and possible improvements

In conclusion, we can say that with our exploration we found out that the GLM Advancements can bring an improvement in performance compared to basic GLM implementations, still maintaining all the benefits of high interpretability and high control of the effects of the variables.

One interesting improvement of the Elastic Net models would be to consider different penalizations for the different variables as described in section 2.1.4. This approach

not only can improve the performance of the models, but would also allow us to systematically introduce prior information in a Bayesian fashion and better control the variables effects.

Another improvement would be exploring the interaction effects in the GLM-based models. On this topic, the GBM fitting could be a nice starting point to get some insights on the interactions and guide the manual introduction of interaction terms in the GLM-based models. GBM models are particularly convenient for these kind of tasks because they mostly work automatically and the manual intervention is minimal, so they can be launched without too much human effort.

Considering the GAM, to better understand their potential, further explorations should be conducted. We hope for the GAM development in H2O to progress. Otherwise we can still try other implementations.

A really important topic in modeling that has not been faced in this analysis and deserves a separate dissertation is the geographical modeling. The geographical modeling is really important in Non-Life Insurance Pricing and requires specific modeling tools that haven't be discussed in this thesis. We remind that in our analysis the choice of using policies from only one specific province strongly reduces the importance of geographical modeling.

The other aspect of considering policies from only one province is that this choice reduces significantly the size of the dataset. This solution allowed us to work locally on one single PC without going out of memory and with still acceptable execution times, but it is clear that with a whole country usual MTPL dataset working locally in a consumer PC would not be possible. The proper solution would be to adopt a cluster of computers big enough to deal with the whole dataset. As we discussed in section 2.3, the cluster computing offers a highly scalable solution and H2O can be easily exploited in a cluster.

Bibliography

[1] Stato Italiano. “Codice Civile”. 2020.

[2] M. P. Wand C. M. Crainiceanu D. Ruppert. “Bayesian Analysis for Penalized Spline Regression Using WinBUGS”. In: *Journal of Statistical Software* (2005).

[3] M. V. Wüthrich. “Data Analytics for Non-Life Insurance Pricing”. ETH Zurich, 2020.

[4] H2O.ai. “H2O: Scalable Machine Learning Platform”. version 3.30.0.6. 2020.

[5] R Core Team. “R: A Language and Environment for Statistical Computing”. R Foundation for Statistical Computing. Vienna, Austria, 2021.

[6] Hadley Wickham and Garrett Grolemund. “R for data science: import, tidy, transform, visualize, and model data”. " O’Reilly Media, Inc.", 2016.

[7] M. Landry. “Machine Learning with R and H2O”. Mar. 2020.

[8] P. Stetsenko. “Machine Learning with Python and H2O”. Mar. 2020.

[9] M. Malohlava, J. Hava, and N Mehta. “Machine Learning with Sparkling Water: H2O + Spark”. Mar. 2020.

[10] T. Nykodym et al. “Generalized Linear Modeling with H2O”. Mar. 2020.

[11] A. Candel and M. Malohlava. “Gradient Boosted Models”. Mar. 2020.

[12] Patrizia Gigante, Liviana Picech, and Luciano Sigalotti. “La tariffazione nei rami danni con modelli lineari generalizzati”. EUT Edizioni Università di Trieste, 2010.

[13] Leonardo Sica and Patrizia Gigante. “Analisi sull’Utilizzo di Variabili Telematiche nella Tariffazione delle Assicurazioni RCA”. Università degli Studi di Trieste, 2016.

[14] Eduardo García Portugués. “Notes for Predictive Modeling”. Carlos III University of Madrid, 2020.

[15] Gareth James et al. “An introduction to statistical learning”. Vol. 112. Springer, 2013.

[16] Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. “The elements of statistical learning”. Vol. 1. 10. Springer series in statistics New York, 2001.

[17] Anthony Christopher Davison. “Statistical models”. Vol. 11. Cambridge university press, 2003.

[18] David Ruppert, Matt P Wand, and Raymond J Carroll. “Semiparametric regression”. 12. Cambridge university press, 2003.

[19] Ulla Holst et al. “Locally weighted least squares kernel regression and statistical evaluation of LIDAR measurements”. In: *Environmetrics* 7.4 (1996), pp. 401–416.

[20] Andrew Gelman et al. “Bayesian data analysis”. CRC press, 2013.