Mémoire présenté devant l'ENSAE Paris
pour l'obtention du diplôme de la filière Actuariat
et l'admission à l'Institut des Actuaires
le 17/03/2022

Par : **Louise Ferbach**
Titre : **Sequential Mortality Modeling with Deep Recurrent Models**

Confidentialité : ☒ NON ☐ OUI (Durée : ☐ 1 an ☐ 2 ans)

Les signataires s'engagent à respecter la confidentialité indiquée ci-dessus

Membres présents du jury de la filière
Nom : Christian-Yann Robert

Membres présents du jury de l'Institut des Actuaires

Entreprise : SCOR
Signature :

Directeur du mémoire en entreprise :

Nom : A. Ly, A. Chancel
Signature :

Autorisation de publication et de mise en ligne sur un site de diffusion de documents actuariels
(après expiration de l'éventuel délai de confidentialité)

Signature du responsable entreprise

Secrétariat:

Signature du candidat

Louise FERBACH

Bibliothèque:

Ecole Nationale de la Statistique et de l'Administration Economique (ENSAE)
5, avenue Henry Le Chatelier - 91120 PALAISEAU, FRANCE

Résumé

Ces dernières années ont vu une accélération croissante de la transformation numérique de l'assurance, qui s'explique par plusieurs facteurs.

Tout d'abord, la prise de conscience par les dirigeants d'entreprises de l'importance des données pour évaluer les risques de décès, dépendance, emprunteur, ... D'autre part, le besoin d'infrastructure performante pour stocker, analyser, et gouverner des données sensibles (contenant des informations personnelles) toujours plus volumineuses. Enfin, la Covid-19 a été un facteur de prise de conscience très puissant et a fortement incité les entreprises à massivement investir dans l'informatique pour rendre le télétravail possible, phénomène qui en s'étendant à la télémédecine rend la collecte de données de santé numériques possible à une bien plus large échelle qu'on ne le pensait jusqu'alors.

Ces changements ont permis aux départements de gestion des sinistres et de souscription de profondément repenser leurs systèmes en automatisant les tâches à faible valeur ajoutée pour pouvoir se concentrer sur l'innovation : nouveau produits d'assurance, partenariat avec des insurtechs et hôpitaux, services intelligents aux clients, ...

Dans un monde numérique collectant toujours plus de données (voitures connectées, smart watches, ...), les assureurs doivent se préparer à comprendre ces nouveaux flux de données pour mesurer les risques et les prévenir. Ce mémoire propose donc une construction de simulateur de séries temporelles biométriques (indice de masse corporelle, tension artérielle systolique et diastolique) impactant le risque instantané de mortalité.

A partir des données générées, l'étude montre que de simples modèles de machine learning adaptés à la modélisation de la mortalité donnent des résultats satisfaisants. Cependant, ces modèles de classification reposent sur une hypothèse forte d'indépendance entre les observations enregistrées à chaque pas de temps.

Ce mémoire présente l'architecture d'un modèle de *deep learning* récurrent ainsi que son implémentation en Pytorch, une librairie Python pour le deep learning. Ces modèles séquentiels sont utiles pour des équipes de tarification afin de mesurer des sensibilités du prix d'une couverture à l'évolution future du mode de vie d'un assuré (alimentation, activité physique, stress, ...).

Ce mémoire d'actuaire orienté recherche apporte un nouveau regard sur la modélisation de la survie grâce aux réseaux récurrents appliqués à l'évolution du profil biométrique d'un assuré. L'étude présente des résultats innovants de tarification adaptative au fil de l'évolution de données biomédicales, plus proches des mortalités réelles que les modèles statiques traditionnels. D'autre part, les modèles de mortalité présentés mesurent seulement la mortalité instantannée en fonction de variables biométriques mais ne possèdent pas de capacité prédictive sur les scénarios biométriques en eux-mêmes. La modélisation des séries temporelles des covariables pourrait être une solution pour prédire un scénario probable d'évolution de la santé de l'assuré à partir d'événements disponibles pour le tarificateur.

Mots Clés: *Machine Learning*, *Deep Learning*, Assurance-vie, Mortalité

Abstract

Recent years have seen an increasing acceleration in the digital transformation of insurance, which can be explained by several factors.

Firstly, the awareness by business leaders of the importance of data to assess the risks of death, dependency, borrower, ... Secondly, the need for an efficient infrastructure to store, analyze, and govern sensitive data (containing personal information) that is increasingly voluminous. Finally, the Covid-19 has been a very powerful awareness factor and has strongly encouraged companies to massively invest in IT to make teleworking possible, a phenomenon that by extending to telemedicine makes digital health data collection possible on a much larger scale than previously thought.

These changes have allowed claims and underwriting departments to radically rethink their systems by automating low-value tasks so they can focus on innovation: new insurance products, partnerships with insurtechs and hospitals, intelligent customer services, ...

In a digital world collecting more and more data (connected cars, smart watches...), insurers must prepare themselves to understand these new data flows to measure and prevent risks. This thesis therefore proposes a construction of a biometric time series simulator (body mass index, systolic and diastolic blood pressure) impacting the instantaneous risk of mortality.

From the generated data, the study shows that simple machine learning models adapted to mortality modeling give satisfactory results. However, these classification models are based on a strong assumption of independence between the observations recorded at each time step.

This paper presents the architecture of a recurrent machine learning model and its implementation in Pytorch, a Python framework for deep learning. These sequential models are useful for underwriting teams in order to measure the sensitivities of the price of a coverage to the future evolution of the insured's lifestyle (diet, physical activity, stress, ...).

This research-oriented actuary's thesis brings a new perspective on survival modeling thanks to recurrent networks applied to the evolution of an insured's biometric profile.

The study presents innovative results of adaptive pricing over the evolution of biomedical data, closer to real mortality than traditional static models. On the other hand, the mortality models presented only measure instantaneous mortality as a function of biometric variables but do not have predictive capacity on the biometric scenarios themselves. Time series modeling of covariates could be a solution to predict a probable scenario of the insured's health evolution from events available to the underwriter.

Key Words: *Machine Learning*, *Deep Learning*, Life insurance, Mortality

Note de Synthèse

En assurance-vie, les actuaires ont besoin d'estimer la probabilité de survie d'un assuré (habituellement par période annuelle, mais aussi trimestrielle ou à toute autre fréquence) pour estimer le risque auquel ils s'exposent et proposer un tarif de prime (Best Estimate) reflétant ce risque aussi précisément que possible.

L'approche traditionnelle est la construction de tables de mortalités par âge, duration, sexe et usage du tabac. L'inclusion d'un plus grand nombre de variables biomédicales (Indice de Masse Corporelle, pression artérielle, ...) ou liées à l'activité physique (rythme cardiaque, nombre de pas quotidiens, ...) dans les systèmes de souscription ont nécessité l'utilisation de modèles statistiques paramétriques (régression de Cox) puis non-paramétriques (Cox-xgboost, forêts aléatoires de survie) pour évaluer le risque de mortalité du client.

L'objectif principal de ce mémoire est d'étudier et modéliser des séquences d'événements ayant un impact sur la mortalité d'un assuré. L'état de santé de l'assuré peut subir des changements importants au cours du temps : perte de poids, stress lié a un changement de situation professionelle, cholestérol élevé associé à une alimentation trop riche, etc. Pour ce faire, il est essentiel de s'interesser aux formes temporelles des observations biomédicales et mesurer leur impact.

Les réseaux de neurones récurrents ont été récemment introduits pour étudier des phénomènes séquentiels, notemment pour l'analyse des séries temporelles ou pour le traitement du langage naturel. Plus précisément, les réseaux de mémoire à long terme (long short-term memory, LSTM) démontrent une capacité à propager des informations passées sur des pas de temps futurs.

Ce mémoire porte sur l'implémentation dans un cadre assurantiel du modèle RNN-Surv introduit dans l'article *RNN-SURV : A Deep Recurrent Model for Survival Analysis* de Giunchiglia et al. paru en 2018. Le modèle calcule d'abord à chaque pas de temps une représentation cachée (embeddings), de dimension inférieure, des variables en utilisant des couches denses. Ce vecteur de caractéristiques est ensuite introduit dans un réseau de mémoire à long terme, qui produit une estimation de la valeur de la fonction de survie à chaque pas de temps en fonction des données actuelles et passées. Enfin, les estimations des probabilites de survie a chaque pas de temps subissent une transformation lineaire produisant un score de risque utilisé comme estimateur de risque pour l'individu.

Ce modèle a la spécificité de fournir deux sorties, qui sont optimisées dans la phase d'apprentissage à l'aide de deux fonctions de perte linéairement combinées. Cette configuration rappelle la structure de l'apprentissage multi-tâche, qui a montré qu'optimiser des multiples tâches annexes, indirectement corrélées à l'objectif principal, améliorait en fait les performances sur chacune des tâches individuelles. De même, ici, nous ne nous concentrons pas sur le calcul d'un score de risque mais plutôt sur les probabilités de survie, ce score de risque (optimisé à l'aide d'une limite supérieure de l'indice de concordance de Harrell) garantit que les individus présentant différents niveaux de risque soient correctement ordonnés par le modèle, tandis que les probabilités de survie réelles sont optimisées à l'aide d'une perte d'entropie croisée (binary cross-entropy loss), modifiée pour prendre en compte la censure.

Une limite du modèle actuel est qu'il n'infére pas l'évolution des covariables médicales à des pas de temps où elles ne sont pas observées (très souvent, les suivis médicaux sont irréguliers). Cependant, le modèle donne de bons résultats par rapport aux méthodes statiques traditionnelles.

Executive summary

In life insurance, actuaries need to estimate the probability of survival of a policyholder (usually on an annual basis, but also quarterly or at any other frequency) in order to estimate the risk to which they are exposed and to propose a premium rate (Best Estimate) reflecting this risk as accurately as possible.

The traditional approach is to construct mortality tables by age, duration, gender and tobacco use. The inclusion of more biomedical variables (Body Mass Index, blood pressure, ...) or physical activity variables (heart rate, number of daily steps, ...) in the underwriting systems required the use of parametric (Cox regression) and then non-parametric (Cox-xgboost, randomized survival forests) statistical models to assess the client's mortality risk.

The main objective of this thesis is to study and model sequences of events that have an impact on the predicted mortality of an insured. The health status of the insured may undergo significant changes over time: weight loss, stress related to a change in professional or social situation, high cholesterol associated with a rich diet, etc. To do so, it is essential to take into account the evolution of biomedical parameters of interest as well as the speed of this evolution and their possible correlation with other concomitant changes: weight gain of 20kg over one year or one month, associated or not with a rise in cholesterol or blood pressure for example.

Recurrent neural networks have recently been introduced in the study of all sequential phenomena, from time series analysis to natural language processing. More specifically, long short-term memory (LSTM) networks demonstrate an ability to propagate past information over future time steps.

This dissertation focuses on the implementation in an insurance framework of the model proposed in the 2018 paper *RNN-SURV: A Deep Recurrent Model for Survival Analysis* by Giunchiglia et al. The model first computes at each time step a lower-dimensional hidden representation (embeddings) of the covariates of interest using dense layers. This feature vector is then fed into a long-term memory network, which produces an estimate of the survival function value at each time step based on current and past data. Finally, the survival function values are linearly combined to compute a unique individual-specific risk score, which is used to obtain a risk ranking of the insured.

This model has the specificity of providing two outputs, which are optimized in the learning phase using two linearly combined loss functions. This setup is reminiscent of the structure of multi-task learning, which has shown that optimizing multiple ancillary tasks, indirectly correlated to the main objective, actually improves performance on each of the individual tasks. Similarly, here we do not focus on calculating a risk score but rather on survival probabilities, this risk score (optimized using an upper bound on the Harrell concordance index) ensures that individuals with different levels of risk are correctly ordered by the model, while the actual survival probabilities are optimized using a binary cross-entropy loss, modified to take censoring into account.

A limitation of the model is that it theoretically provides no explicit constraint on the survival probabilities which must be strictly decreasing, although in practice we observe that the model mostly captures this phenomenon and produces decreasing survival probability estimates. A variant that predicts the hazard rate may be an alternative, but it has not shown any significant improvement in predictive performance. Similarly, there are different ways to infer the evolution of medical covariates at time steps where they are not observed (very often, medical follow-ups are irregular). However, the model performs well compared to traditional static methods.

Remerciements

Je tiens à remercier SCOR Global Life pour m'avoir donné l'opportunité de réaliser cette étude passionnante et enrichissante.

Je tiens à remercier Antoine Ly pour ses conseils et son suivi tout au long de ce mémoire, Christian-Yann Robert pour ses recommandations de modélisation biométrique, Roberto Castellini pour ses remarques judicieuses sur les différentes méthodes de deep learning envisagées.

Enfin, mes remerciements vont tout particulièrement à Antoine Chancel, sans qui ce mémoire n'aurait jamais vu le jour ni n'aurait atteint sa qualité de contenu, pour son suivi dévoué tout au long du stage, ses remarques judicieuses, sa présence, ses explications et surtout ses idées toutes plus pertinentes les unes que les autres à explorer.

Contents

Resume **3**

Abstract **5**

Note de Synthèse **7**

Executive summary **9**

1 Study framework **14**

1.1 The life insurance industry . . . 14

1.1.1 Main departments and roles in an insurance company . . . 15

1.1.2 New Data and Ethics Challenges . . . 16

1.2 The Reinsurance Industry . . . 17

1.2.1 Presentation . . . 17

1.2.2 Types of Reinsurance . . . 19

1.3 Reflexions on event-based life insurance . . . 19

1.3.1 Insurance context . . . 19

1.3.2 Medical follow-up of a patient . . . 20

1.3.3 Continuous Underwriting . . . 23

1.3.4 Pricing . . . 24

1.3.5 Capital requirement . . . 25

1.3.6 Prevention care . . . 25

2 Actuarial Science Tooling **26**

2.1 Survival Analysis Theory . . . 26

2.1.1 Survival Analysis in life insurance . . . 26

2.1.2 Censoring and truncation . . . 28

2.1.3 Quantity of interest . . . 31

2.2 Exposure derivation . . . 33

2.2.1 Initial Exposure and Balducci hypothesis . . . 34

2.2.2 Central Exposure and Constant hazard function . . . 35

2.2.3 Modeling our data using exposure . . . 37

2.3 Mortality modeling . . . 37

2.3.1 Non-parametric estimators 37
2.3.2 Parametric extimators 40
2.4 Machine learning models in survival analysis 46
2.4.1 Cox-Model adaptations 47
2.4.2 Survival Trees 49

3 Data Generation 54
3.1 Assumptions and key criteria 54
3.2 Data generator schema 55
3.2.1 From an initial distribution... 55
3.2.2 ...To a dynamic evolution... 58
3.2.3 ...Ending in Death or Censorship 60
3.3 Descriptive statistics of the dataset 62
3.3.1 Initial distribution 62
3.3.2 Entire dataset 64
3.3.3 Limitations 67

4 Deep Learning in Survival Analysis 69
4.1 Deep Learning in a word 69
4.1.1 Introduction 69
4.1.2 Principles 70
4.1.3 Gradient Descent and Back-Propagation 72
4.1.4 The Adam Optimizer 74
4.2 Deep Learning for Sequential Data 76
4.2.1 Recurrent Neural Networks 76
4.2.2 Long Short-Term Memory 78
4.3 Survival deep learning 79
4.3.1 Cox Proportional Hazards Models 80
4.3.2 Static Architectures 81
4.3.3 Recurrent Deep Learning Models 83
4.4 RNN-SURV: a Deep Recurrent Model for Survival Analysis 84
4.4.1 Model 84
4.4.2 The Concordance Index 87
4.4.3 Censored Cross-Entropy Loss 93
4.4.4 Optimization 94
4.4.5 Hyper-parameters cross-validation 100

5 Pricing applications and scenarii 101
5.1 Assumptions 101

5.2 Models Training and Testing 102
5.2.1 Machine Learning Models 102
5.2.2 RNN-Surv 103
5.3 Models Evaluation 105
5.3.1 Scenario Formulation 105
5.3.2 Payments 105
5.3.3 Central health scenario 106
5.4 Increasing BMI scenario 108
5.5 Decreasing BMI scenario 109

Conclusion **111**

Bibliography **118**

Chapter 1

Study framework

What is an Actuary ?

According to Investopedia[1], "an actuary uses math and statistics to estimate the financial impact of uncertainty and help clients minimize risk." His activities cover a large panel of statistical tooling, like calculating the Best Estimate of a risk (as in life insurance for example) or trying to market new emerging risks (cybersecurity or terrorism for example). He can be found in an insurance or reinsurance company as well as in a traditional financial department (of a bank for example), an insurtech or fintech, or a research and development department. Very often, his work involves dealing with regulatory compliance issues such as the European legislation Solvency 2, or accounting responsibilities such as calculating the actuarial reserves in order to face upcoming risks.

1.1 The life insurance industry

Life insurance is a contract between a policy holder and an insurance company, where the insurer commits to paying a designated beneficiary (wife, children, charity...) a fixed or variable sum of money upon the death of an insured person (very often the policy holder). Depending on the contract, other events such as critical illness can also trigger payment. The policy holder typically pays a premium, either regularly or as one lump sum. The nominal can be a predefinite amount (e.g. $1million), depending on conditions (e.g. doubled in case of accident) or variable (e.g. a fixed number of stocks from a certain company, without prior knowledge of the value these stocks will bear at the death of the insured).

The first life insurance policy was proposed by the *Amicable Society for a Perpetual Assurance Office* founded in London in 1706 by William Talbot and Sir Thomas Allen,

---

[1] https://www.investopedia.com/articles/professionals/090513/day-life-actuary.asp

and initially recorded about 2,000 members. Each member between the ages of twelve to fifty-five would pay a fixed annual payment (premium), in exchange for earning shares, from one up to three depending on the member's age. At the end of the year a portion of dividends was divided among the wives and children of the deceased members, proportionally to the shares owned by the deceased members.

The life industry became more complex as the modern life insurance industry offers now all kinds of products covering death events or other hazards related to the health condition of the insured. The main products are covering **Mortality**, **Critical Illness**, **Medical Expenses**, and **Longevity** (Pension).

Modern life insurance bears some similarity to the asset-management industry, and life insurers have diversified their product offerings into retirement products such as annuities. Life insurance policies can be divided into two main categories :

- **Protection policies:** designed to provide a benefit, typically a lump-sum payment, in the event of a specified occurrence, often the death of the policy-holder.
- **Investment policies:** these policies can be seen as financial instruments, their main purpose is to increase a capital constituted by regular or single premiums and on which some interest is calculated.

1.1.1 Main departments and roles in an insurance company

The life insurance industry collects insurance premiums and invests in assets globally that can result in systemic risk for the economy. The recent regulations, such as Solvency II, force insurers, and reinsurers to increase their capital requirements and to increase their knowledge of the risks to reduce the chances of bankruptcy. In this spirit, the insurers' structure has been rethought. To ensure the economic prosperity of the company, four main departments described below work closely together.

Underwriting department role is to provide acceptance or rejection of an insurance application. To do so, underwriters assess the risk based on the information collected from the life insurance applicant. If the risk appears to be high but acceptable, an extra premium may be demanded. This department has a key role as it avoids the insurer to be anti-selected and insured the risk covered corresponds to premiums collected.
To grow its market share, an insurer endeavors to reduce the complexity of the underwriting process. Accessing information from a client is expensive. This cost includes the easiness to access to information. For example, asking for the age of a person is easier than collecting blood measurements. On the other hand, an additional accurate piece of information improves insurer knowledge of the risk.

The **Experience Analysis (EA)** department is in charge of monitoring the in-force portfolio. By comparing the observed claims with the expected ones, the team report about the health of the business. The team also provides insights into the assumptions that should be revised to better reflect the risk. More generally, the Experience Analysis team provides data and insights about the risk assumptions considered in pricing and valuation.

The **Research & Development** department is providing support to other teams for risk assessment. The team provides guidance and best practices to set assumptions for various products and risks. When data is not available internally to assess the risk, the R&D team often formulate recommendations based on external literature and data sources. In particular, R&D is responsible to estimate future trends and estimation of rare events (1-in-200 years scenarios).

The **Pricing** department works closely with the EA and R&D to set an adequate price for the insurance products. The pricing team relies on EA and R&D inputs on risk to forecast future claims. Considering other economic and financial assumptions the team determines the premiums need to meet the profitability targets. As the price competition between insurers or reinsurers is high, the pricing teams work closely with the underwriting teams to make sure that the underwritten business is profitable.

1.1.2 New Data and Ethics Challenges

Thanks to the technological improvement and data storage capacity, information considered for risk assessment increased a lot in recent years. For instance, nowadays life insurance applicants are required to share, besides their age, a significant part of their medical history, financial situation, profession, etc. As the available information increases, new challenges for insurance companies and actuaries are arising, namely extracting and analyzing efficiently information from a large amount of (often unstructured) data to better assess the risk.

In the same view, the use of big data raises growing ethics concerns over discrimination possibilities : for example, the intake of the profession or home address could be considered discriminatory against some minorities that are widely dominant across harder jobs, for example with frequent exposure to carcinogenic chemicals (construction workers...), or living in riskier neighborhoods or with less access to quality food supplies.

In a different perspective, someone who experienced serious health problems during his childhood or early youth could find himself being discriminated against at subscription of an insurance contract for his whole life, his premium being weighted by that past health hazard. In the European Union, important legislation exists (e.g. AERAS in France) to guarantee a *right to be forgotten* in such situations : for a cancer before 21, after 5 years without relapse, and for a cancer after 21, after 10 years without relapse. It has recently been proposed to lower the timeline at 5 years without relapse for all population.

1.2 The Reinsurance Industry

1.2.1 Presentation

Reinsurance is a coverage that an insurance company purchases from another one (either a specialist reinsurance company, which only undertakes reinsurance business or another insurance company) to protect itself, partially or totally, from the risk of a major claim events. For example, an insurance company may provide home insurance in case of a plumbing-induced water damage, but might be unable to cover simultaneous claims for 100 houses in the same neighborhood due to a major flooding event. With reinsurance, the company passes on ("cedes") some part of its own insurance liabilities to the reinsurer. The company that purchases the reinsurance policy is called a "ceding company" or "cedant".

The main purpose of reinsurance is to allow insurance companies to remain solvent after major claim events, such as major disasters like hurricanes and wildfires, or a mortality peak due to a major epidemic. Almost all insurance companies have a reinsurance program to reduce their loss exposure : the insurer can then constitute bigger portfolios than would otherwise be allowed by the regulator, being able to take on more risk because some of it is now transferred to the reinsurer. This is likely to reduce the amount of capital needed to provide coverage. Therefore, in addition to its basic role in risk management, reinsurance is used to reduce the cedant's capital requirements.

Reinsurance can make an insurance company's results more predictable by absorbing large (and rare) losses, thus truncating the tail of the loss distribution. The risks are spread, with the reinsurer or reinsurers bearing some of the loss incurred by the insurance company. The income smoothing arises because the losses of the cedant are limited and highly predictable (it is easier to forecast results regarding high-frequency, low-cost events than low-frequency, high-cost events). This fosters stability in claim payouts and caps indemnification costs.

The focus of reinsurance on rare risks or extreme events gives the reinsurer an expertise that the insurer may want to use for setting an appropriate premium, in regard to a specific risk or when launching a new product with no known performance history. The reinsurer would also prefer to apply this expertise to the underwriting process in order to protect its own interests. By choosing a particular type of reinsurance method, the insurance company may be able to create a more balanced and homogeneous portfolio of insured risks. This would make its results more predictable on a net basis (i.e. allowing for the reinsurance). This is usually one of the objectives of reinsurance arrangements for the insurance companies.

The insurance company may be motivated by arbitrage in purchasing reinsurance coverage at a lower rate than they charge the insured for the underlying risk, whatever the class of insurance. The reinsurer may be able to cover the risk at a lower premium than the insurer :

- The reinsurer may have a greater risk appetite than the insurer.
- The reinsurer may have some intrinsic cost advantage due to economies of scale or some other efficiency.
- Reinsurers may operate under weaker regulation than their clients. This enables them to use less capital to cover any risk, and to make less conservative assumptions when valuing the risk. Even if the regulatory standards are the same, the reinsurer may be able to hold smaller actuarial reserves than the cedant if it thinks the premiums charged by the cedant are excessively conservative.
- Reinsurers will often have better access to underwriting expertise and to claims experience data on extreme events than insurers have, enabling them to assess the risk more accurately and reduce the need for contingency margins in pricing the risk.
- The reinsurer may have a more diverse portfolio of assets and especially liabilities than the cedant. This may create opportunities for hedging and lower the capital requirements.

To mitigate the counterparty risk, many reinsurance placements are shared between a number of reinsurers. Using game-theoretic modeling, Professors Michael R. Powers (Temple University) and Martin Shubik (Yale University) have argued that the number of active reinsurers in a given national market should be approximately equal to the square-root of the number of primary insurers active in the same market. Econometric analysis has provided empirical support for the Powers-Shubik rule.

1.2.2 Types of Reinsurance

There are two methods of reinsurance:

- **Facultative Reinsurance**, which is negotiated separately for each policy. It is normally purchased as complement for individual risks not covered by other reinsurance treaties, for amounts in excess of the monetary limits and for unusual risks. Underwriting expenses are higher for such business because each risk is individually underwritten and administered ; furthermore, risk mitigation is difficult in this setup. However, as the reinsurer can separately evaluate each risk they take on, the contract can be priced more accurately.

- **Treaty Reinsurance** means that the cedant and the reinsurer enter a reinsurance contract which covers a specified share of all insurance policies issued by the cedant. The reinsurance treaty may force the reinsurer to accept all policies within the scope ("obligatory" reinsurance), or it may allow the insurer to choose which risks it wants to cede, with the reinsurer obligated to accept them ("facultative-obligatory" reinsurance). There are two types of treaty reinsurance :

  - Under **proportional reinsurance**, the reinsurer's share of the risk is applicable to each individual policy ; he takes on a share of each policy's risk, in exchange of the same share of premiums. In addition, he might allow a "ceding commission" to the insurer to cover the uderwriting costs. Proportional reinsurance may be *quota share* (a fixed percentage of each policy) or *surplus reinsurance* (the cedant retains the full amount of each risk up to a maximum called "retention limit", and the excess loss is reinsured).

  - Under **non-proportional reinsurance** the reinsurer's liability is based on the aggregated claims ; he pays out if the total claims suffered by the insurer in a given period exceed a stated amount ("retention" or "priority"). The main forms of non-proportional reinsurance are excess of loss and stop loss.

1.3 Reflexions on event-based life insurance

1.3.1 Insurance context

The life insurance industry was created to cover risks associated to the life of individuals. The insurer promise the policy holder that one or many beneficiaries will receive a payment based on an event (ie. death, critical illness). Estimating the price of the future promise is a difficult exercise that requires the knowledge of the future evolution of the characteristics of the individual at risk. To tackle this issue, life actuaries and statisticians developed many techniques to estimate the survival probability of a new policy-holder based on the historical data collected by the insurance company. The estimated survival probability help life insurers to have an indication the price of the policy and the reserves attached to the contract.

Traditionally, life insurance companies know a client based on an online or hand written application. During the underwriting stage, more information can be collected : pharmaceutical history, client interview, MIB, Laboratory report ... Over the past decade, life actuaries endeavored to deal with insurance and underwriting portfolio with more individual features and more applications / policies. This inflation of data leads to numerous issues with parametric techniques : complex non linear relationships between the features, gradient descent may not work in the context of big data (inversion of very large matrices), ...

During the past 5 years, data streams are emerging with the apparition of the internet of things and the improvements of health care IT systems. The connected objects collect trough wearable (smart watch, smart pants, smart glasses, ...) real time data : heart pulse, distance, steps number, calories burnt, ... On the other hand, some insurtechs develop new electronic system that are able to centralize medical reports of many GP in a single application. Partnering with those new tech companies, the insurance industry is also able to monitor the physical activity, the quality of sleep or the medical condition of a client through the lifetime of the policy. Understanding the improvements or deterioration of the policy holder could an insurer to prevent any unfortunate event. Also, the insurer may propose new insurance coverage with discounted premium if the insured health condition improve. Based on the seriatim data observed for a particular individual, the insurance company need to project the best estimate of the future evolution of the covariates and the related mortality predicted by the model every year.

1.3.2 Medical follow-up of a patient

Nowadays, health status data is becoming more numerous and increasingly available from a variety of sources: medical reports issued by a general practitioner, number of steps per day and heart beat collected with a smart watch, screen time measured with phone or computer, sleep analyzer, etc.

Life insurers traditionally request applicants to fill a medical questionnaire including

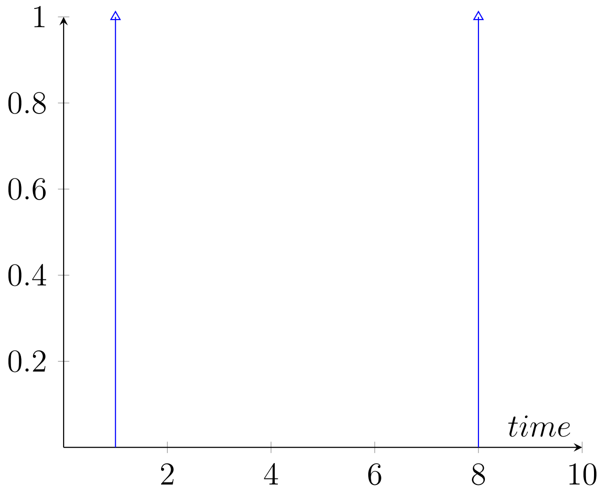


Figure 1.1: Events

certain medical history. Life underwriters can also gather more information : MIB (Motor Insurers' Bureau) records, medical exam, blood or urinary samples analysis, pharmaceutical history, social media posting, or any other evidence to avoid fraud or disclosure. The insurer can also monitor a change of marital status through a mobile or a web application. Later, a claim may be submitted to the insurer with an eventual payment. The figure 1.1 represents a sequence of the events.

What is often crucial, medically-speaking, is the *evolution* of given parameters. Here are a few examples.

Smoking Evolution

By incorporating national UK mortality rates for lung cancer in males, the cumulative risk of death from lung cancer by age 75 among current smokers was estimated at 16%, rising to 24% for current smokers of at least 25 cigarettes per day. The benefit of quitting smoking was demonstrated with cumulative risks of lung cancer of 10, 6, 3 and 2% for men who stopped smoking at ages 60, 50, 40 and 30, respectively. It is therefore apparent that smokers who quit, even well into middle age, avoid a large proportion of their subsequent risk of lung cancer.

Therefore, knowing that a policy-holder is a smoker at year k is only a very partial information on his probability to develop lung cancer in 10 or 20 years, it would bring much more information to have regular follow-ups telling us, for example, if the policy-holder keeps smoking his entire life, or quits smoking at year $k+1$, or $k+10$. The impact of the evolution of that habit on the likelihood to develop lung cancer is essential to be incorporated in our survival models, as a static measure is not an accurate-enough information on the probability of lung-cancer-related death.

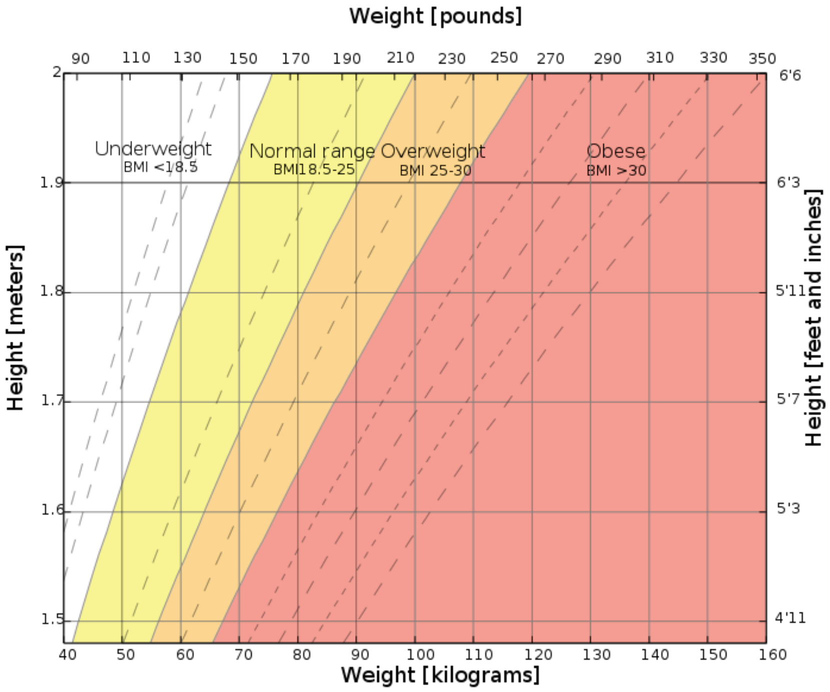


Figure 1.2: Body Mass Index Chart

One can even mention the trickier adverse-selection effect : for long-time smokers, having quitted smoking is not always a good sign, as it is often an indicator that the policy-holder has experienced very severe health issues forcing him to stop smoking. Therefore, capturing the overall impact of the evolution of smoking habits is a delicate matter.

BMI evolution

Another example is the evolution of a person's weight, or his body mass index (BMI), but as the height remains constant once adulthood is reached this is equivalent. The body mass index is a person's weight in kilograms divided by his squared height in meters. BMI screens for weight categories that may lead to health problems, see figure 1.2. Even more than with smoking habits, the *evolution* of the BMI can be a very good indicator of a policy-holder's health improvement or deterioration.

Let's take, for example, an adult male of 180cm, weighting 120kg on January 1^{st}, year k. He has a BMI of 37, and is in type II obesity (see figure 1.2), therefore is at higher risk of stroke, coronary disease, heart disease and a lot of other health issues than the general population. Were his BMI to increase in any proportion, it could probably only have negative effects on his long-term health prospects. Now, if after great effort, 1 year later on January 1^{st}, year $k+1$ he only weighs 90kg, his BMI has fallen at 27.8 and he is back in the slightly overweight, non-obesity range : that is probably very positive for his survival probability.

But what if this 30kg weight loss had occurred in 1 month rather than 1 year, would it be a good sign ? It would much more likely be an indicator that the policy-holder is experiencing serious health problems (cancer, depression...), or at least that this massive weight loss must have induced severe nutritional deficiencies. Similarly, if after taking 1 year to fall at a BMI of 27.8, it takes him just one month to climb back at the initial level (or even higher), it is not at all as good a sign for his life expectancy compared to having maintained his weight loss.

Therefore, BMI is a very complex indicator of the overall health status of an individual, and its evolution, though complex to analyze, would provide much better insight in the survival probability.

To take into account the complexity of the relationship between the evolution of some medical features and the short-, mid- or long-term survival probability of a policy-holder, some insurance companies propose *bonus-malus* programs similar to those in car insurance, based on the policy-holder medical, dietary and physical habits, e.g. you could have a minoration of your insurance premium if you walk an average of more than 10,000 steps a day, a majoration if you start to smoke, or for over-70 you could have a yearly bonus for getting the flu shot, or the covid vaccine, etc.

1.3.3 Continuous Underwriting

Life insurance providers are constantly trying to improve the underwriting process to reduce policy risks. Automated rule engines have increased the accuracy and speed of underwriting for a specific point in time, but providers are still essentially making educated guesses on how a person will age for the next fifty years. Furthermore, that assumption is subjected only to the perceived health condition at time of underwriting, and the insured could modify his habits for the better or the worse, what would affect his survival probabilities. Example : he could start / quit smoking ; change his nutritious habits ; move to a more/less polluted area...

Continuous underwriting means that data can be analyzed before, during and after the policy is underwritten, to minimize insurers' risk by a dynamic adaptation of premiums to the insured's behavior, measured by the tremendous amounts of customer data generated by the Internet of Things (IoT)/wearable technology, customer engagement platforms and social media.

Life insurers would be able to provide a comprehensive, individualized risk assessment over time, rather than relying on small and incomplete sets of data to guess how an individual will age. Utilizing this data, insurers can reward healthy behavior, like quitting smoking, getting vaccinated or walking more than 10,000 steps a day, and build a richer underwriting experience that can be fed back into new products being developed.

Continuous underwriting would introduce dynamic risk selection, and provide an incentive for insureds to take care of their health. To that purpose, the contract would specify that the data can be collected from the insured and through which channel (e.g. a connected watch measuring sleep time, or a phone application counting daily steps, ...) in exchange for lower premiums.

1.3.4 Pricing

We would like our pricing strategy to be reflective of the insured's initial survival probabilities, but also to be updated depending on an eventual change of health condition. More explicitly, if the insured respects a good lifestyle, we would like the new price to be reflective both of all past events of the insured but also of his updated survival probabilities given the change in his habits.

Example : at time of underwriting, a 40-year-old man has a BMI of 37 (type II obesity). This corresponds to a weight of 120kg for a 1.80m tall person. The initial estimation of his risk is driven by obesity-induced comorbidity (higher cardiovascular risk among others). His premium is therefore higher than it would have been with a BMI of 27 (equivalent to 87.5 kg).

Now, this same person goes a diet and starts practicing a regular physical activity. Over the following year, he loses weight steadily until his BMI stabilizes at around 27 (slightly overweight but below the type I obesity threshold of 30). This naturally improves his chances facing most cardiovascular diseases, diabetes, etc. Therefore, we would like to lower his forecast mortality risk and update his premium to be reflective of that improved health condition.

However, if that same weight loss of more than 32kg happened not over a year, but over a month, the conclusion would likely be very different. Indeed, a massive and sudden weight loss is more often the sign of a deterioration of one's health condition (depression, cancer, etc), or would at the very least necessitate such strict fasting that it would cause severe nutritional deficiencies. Therefore, it is essential to update the survival probabilities in a dynamic manner, taking into account the history and timing of medical features evolution, and not only in a static, instantaneous manner.

1.3.5 Capital requirement

In order to determine his capital requirements, the insurer has to provide a best estimate of his liabilities to the regulator. It is also essential for managing the risk the company is undertaking to have a clear evaluation of the necessary capital to face the upcoming claims in the short- and long-term future, to estimate if this risk is tenable and at what maturity to invest the collected premiums.

The evalutation of the required capital can be made based on the events of all the insureds, but it would be relevant for future claims to calculate a projection of the portfolio population lifestyle and likely mortality evolution. For example, relying on past obesity-related comorbidities (diabetes, cardiovascular diseases, etc) to predict the future might be off-target since it is largely documented that obesity is on the rise worldwide, from 6% of the adult population in the early 1980's to 13% in 2016, and this trend is not likely to slow down, let alone reverse. Developed economies, that also happen to be at the core of the life insurance business, are particularly subject to the abundance of low-cost, highly-processed industrial food driving the obesity epidemic.

1.3.6 Prevention care

Integrating a prevention approach in continuous underwriting would help early health risk detection, thus orienting care and efforts very early and avoiding a lot of the costly and much more hazardous long-term consequences of a lack of screening or prevention. What's more, connecting insureds with a phone application that would encourage healthy eating and regular physical activity by promising premium discounts could be a powerful incentive to change habits towards a healthier global behavior, improving forecast survival probabilities in a very tangible, measurable way through the electronic device.

Chapter 2

Actuarial Science Tooling

2.1 Survival Analysis Theory

2.1.1 Survival Analysis in life insurance

Survival Analysis is a branch of statistics that focuses on analyzing the waiting time before occurrence of an event, typically death but also disease incidence, termination of an insurance contract, recovery, or any designated event of interest that may happen to an individual. The time may be measured on a different scale such as years, months, or days from the beginning of the follow-up of an individual until the event occurs. The time variable is usually referred to as survival time because it gives the time that an individual has "survived" over some follow-up period.

A standard **term life insurance policy** of nominal N is defined by a contract where a subscriber (policy-holder) pays a premium π, in exchange of an amount N shall be paid to a beneficiary at occurrence of an event, very often the death of the policy-holder. The main purpose of life insurance is to estimate, as accurately as possible, the optimal value of the premium π depending on a number of medical information regarding the subscriber : age, personal and family medical history, body mass index (BMI), average daily sport activity... Therefore, modeling biometric risk, i.e. modeling the duration between the application submission date and claim occurence, is essential in the life insurance industry as it impacts pricing, reserving, and solvency assessments.

Several variants of this standard contract exist, where the trigger element is not death but occurrence of a severe illness or accident (e.g. cancer), but the general idea being the same we will focus on the study of lifespan. The three main durations of interest in the life insurance industry are :

- Life span, the duration before death of the insured,
- Disability duration, the period an insured will remain disabled (before death or

recovery),

- Autonomy (not disabled) duration, the duration before either death or loss of autonomy.

For the computation of premiums and reserves, one may not limit the modeling to a binary classification problem (claim occurrence). Indeed, it is important to estimate the event occurrence probability at any time for the whole duration of the contract : in other words, we seek to predict the claim probability for a given period of time, which is the reason why it is important to master survival analysis theory.

Another important reason why we may wish to estimate claim occurrence probabilities as exactly as possible is to give accurate estimates to the underwriting department, that needs to classify insurance applicants in *bad* or *good* risk categories on their risk profile, meaning being able to order applicants based on their probability of claim occurrence. Besides, the development of more precise models contributes to being more inclusive, as we may derive a price even for a very risky individual. Thus it enables to sell products to clients, who would have been excluded from the portfolio in the past.

Life insurers need to estimate life duration to bring out risk factors. Predicting time to event requires a specific modeling approach called *Survival Analysis*. However, survival data is particularly difficult to collect : contrarily to instantaneous data, such as weight or age, that can easily be obtained, the constitution of a survival database faces an important issue, which is the duration. Very often, survival analysis studies span over years, if not decades ; this leads to facing important issues regarding the integrity of the data, as is explained in the following.

Let's say we have several individuals for whom we want to predict the death date. For example, we want to study the survival of cancer patients from the date of diagnosis. These individuals, in the case of follow-up of a cohort over a long period of time (as is necessary in survival analysis), may be subject to censorship, that is to say, some of them might leave the country, stop showing up at doctors' follow-up appointments, or die of another cause than the one being studied.

For example, among 10,000 cancer patients over a decade, statistically speaking, several might probably move abroad, or one might likely die of a car accident during the study's course (the US car accidents death rate was of 12 per 100,000 in 2019) ; this death would then be treated as censorship since it has on principle nothing to do with the scope of the study. More precisely, it is designated as a competing risk : a risk which has the same outcome as the one under study (death), and therefore prevents the observation of the event of interest.

As survival studies are quite extended in time, typically several years, censorship rates might be high. Finally, one has to end the study at some point : the remaining individuals who are still not dead at that point then get censored.

For censored individuals, we don't have a death date, but we still have an information : we know that they lived *at least until a certain point*, which is better than having no information at all.

It would be foolish to simply discard censored individuals : in the study of cohorts over a very long period, they are likely to represent a consequent proportion, if not the majority, of the subjects, so that would drastically reduce the size of our cohort.

What's even worse, it would introduce a considerable bias towards high mortality: indeed, individuals surviving longer are naturally more likely to be censored at some point (since they are present in the study for longer). Inversely, someone dying right after the beginning of the study would not have had the time for some censorship to happen. In conclusion, discarding censored individuals would lead to overestimating mortality rates, or underestimating survival times.

Similarly, we couldn't either consider censorship dates as death dates : as the individuals are still alive at this point, it would also lead to underestimating survival times, overestimating death rates.

Therefore, it is necessary to use a metric that takes into account censored individuals information as partial, yet significant information, yet not distorting it by granting it the same status as complete survival information from other individuals.

The main specificity of survival data is *censoring*. Indeed, most of the time survival duration is only partially observed. Because of it, different approaches must be considered to include this specificity. Two main modeling strategies exist to take censoring into account: fitting specific models to raw survival data or modifying the data structure to be able to apply standard models.

This chapter will introduce the survival analysis theory: the data challenges and the modeling strategies, on which Machine Learning modeling relies.

2.1.2 Censoring and truncation

A major problem n survival analysis is the fact that the period of observation can be impossible to directly measure for many individuals i. Indeed, survival data by definition necessitates very long observation periods, and by the time death occurs multiple events could have happened that would remove individuals from the observation framework. For example, if we are studying a population of breast cancer patients, since this diagnostic has a rather good prognostic (survival rate of 85% after 5 years), it is very likely that we won't be able to observe all death dates :

- Some people might not be dead by the time we close the study ;
- Some people might die of other causes (suicide, accident, heart attack ...) ;
- Some people might leave the study (move abroad, change doctor...) ;
- IT system failure during records...

These are called *Censoring* and *Truncation*. For such individuals, we only know that they survived for *at least* C_i, i.e., our *actual observation* is

$$(Y_i, \delta_i) = (min(T_i, C_i), \mathbb{1}[T_i \leq C_i])$$

In other words, we don't observe the death date T_i for all individuals, but we observe both when they leave the study Y_i and why ($\delta_i = 1$ if that corresponds to a death, $\delta_i = 0$ if it is a censorship).

Definitions

The main limits to obtaining information in a survival analysis study are censoring and truncation of data.

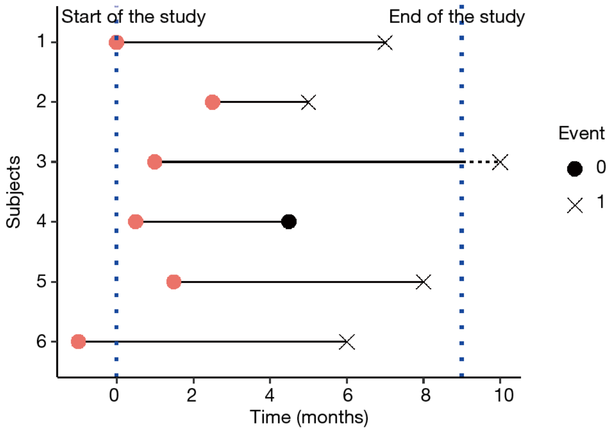


Figure 2.1: Illustration of censoring.

- **Right Censoring**: Right censoring refers to the case when the event of interest either occurs after the end of the observation period or is unobserved due to the loss of the subject due to other independent reasons. In Figure 2.1, subjects 3 and 4 are subject to right censoring. Even if the exact duration is not observed, right censoring still reveals partial information: we have an inferior bound on the survival duration.

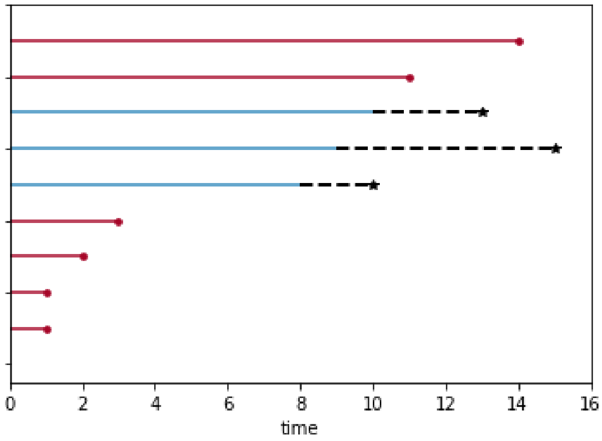
- **Left Censoring** : Left censoring occurs when the trigger point of the duration measure is before the observation period as it is the case for the subject 6 in Figure 2.1. Once again, we only know that the real duration is superior to the one observed.
- **Right Truncation** Right truncation corresponds to individuals who are completely excluded from a study because the starting event that includes them in the study happens after the end of the observation period.
- **Left Truncation** Left truncation is the opposite. An individual is excluded because the event of interest occurs before the beginning of the observation period.

Most of the time, for studying biometric risks in life insurance, left censoring and right truncation do not occur.

Left truncation is more likely to happen when modeling life risks, but in the following, we will only deal with the right censoring, which is the most common scenario. It is worth noting that it is possible and quite easy to consider left truncation by enhancing a bit of the modeling.

Impact on duration estimation

When dealing with survival data, a common mistake could be to simply ignore any censor or truncation effects. This approach leads to an underestimation of the interest event probability. Another common mistake is to restrict the study to only observations that are complete by removing any censored or truncated records. Here as well, estimation is extremely biased. Let consider the ten following individuals to understand the intuition behind the importance of considering censoring:



	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*
Duration observed	10	9	1	3	0	2	8	1	14	11
Real duration	13	15	1	3	0	2	10	1	14	11
Status	C	C	D	D	C	D	C	D	D	D

Figure 2.2: Illustration to highlight the impact of censor

Based on this data, the mean survival duration period is 5.9 when considering censored time as death time and 5.3 when removing censored observation while the real one is 7. In both cases, in this example, life expectancy is underestimated when the partial information coming from the censored individuals is ignored. In other words, the factor of risk will be overestimated.

2.1.3 Quantity of interest

The theory focuses on two functions as the quantity of interest to estimate: the **survival function** S and the **hazard function** h. Having an estimation of one of them allows to fully model the survival of an individual.

The Survival Function

The **survival function** S represents the probability that the time to the event is not earlier than a specific time t:

$$S(t) = Pr(T > t) \tag{2.1}$$

Let us recall that the cumulative distribution function for the random variable T uniquely determines $\mathscr{L}(T)$ the law of T :

$$F(t) = \mathbb{P}(T \leq t)$$

From here, we define the survival function :

$$S(t) = \mathbb{P}(T > t) = 1 - F(t^-)$$

The survival function is decreasing from 1 to 0. The meaning of a probability equals to 1 at the starting time is that 100% of the observed subjects are alive when the study starts: none of the events of interest have occurred. From this quantity, we can define the *cumulative death distribution function* $F(t) = 1 - S(t)$ and the *density function* $f(t) = \frac{dF(t)}{dt} = \frac{-dS(t)}{dt}$ for continuous cases and for discrete cases $f(t) = \frac{[F(t+\Delta t) - F(t)]}{\Delta t}$. The relationship among these functions is shown in Figure 2.3.

However, in circumstances like life insurance where the subscriber already has a certain age τ at time of entry in the portfolio, one is more interested in $\mathscr{L}(T|T > \tau)$ the conditional law of T :

$$S(t|x) = \mathbb{P}(T > t|T > \tau) = \begin{cases} 1 & \text{if } t \leq \tau \\ \frac{S(t)}{S(\tau)} & \text{if } t > \tau \end{cases}$$

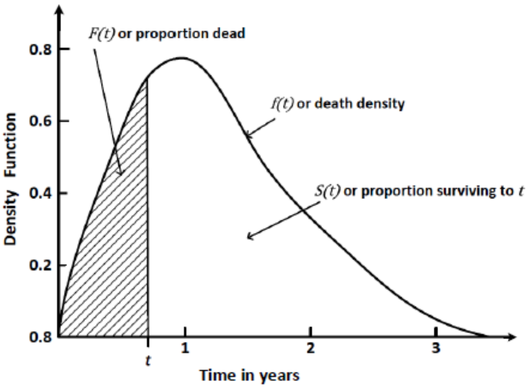


Figure 2.3: Relationship among $f(t)$, $F(t)$ and $S(t)$

The Hazard Rate

In the case of lifetimes, consider an individual is of age t. One wishes to define a quantity that tells us whether this individual is strongly at risk right now, or if the probability that he dies in the near future is low. The hazard rate intends to express the risk that an event (death) occurs in the immediate future, knowing that it did not occur before.

It indicates the rate of event at time t, given that no event occurred before. Formally, the hazard rate function is defined as:

$$\begin{aligned} h(t) &= \lim_{\Delta t \to 0} \frac{Pr(t \leqslant T \leqslant t + \Delta t | T \geqslant t)}{\Delta t} \\ &= \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t S(t)} \\ &= -\frac{\mathrm{d} \log S(t)}{\mathrm{d}\, t} \end{aligned} \tag{2.2}$$

From this equation we can easily derive that

$$S(t) = exp(-\int_0^t h(s)\mathrm{d}s) = exp(-H(t))$$

where $H(t) = -\int_0^t h(s)\mathrm{d}s$ is called *the cumulative hazard function.*
Using the same notation as before, we can define a likelihood function taking into account censoring:

$$L = \prod_i P(T = t_i)^{\delta_i} P(T > t_i)^{1-\delta_i} = \prod_i h(t_i)^{\delta_i} S(t_i) \tag{2.3}$$

The intuition of the function comes from the contribution to the likelihood function between a censored and a full-observed individual:

- If an individual dies at time t_i, its contribution to the likelihood function is indeed the density that can be written as $S(t_i)h(t_i)$.
- If the individual is still alive at t_i, all we know is that the lifetime exceeds t_i, which means that the contribution to the likelihood function is $S(t_i)$.

2.2 Exposure derivation

Time discretization, typically on an annual basis, is a very common way to assess mortality (example : life tables, life expectancies by age...). However, it introduces several challenges with respect to the accuracy of the reported death rate on each of the considered time intervals. Indeed, withdrawing censored subjects from the study introduces bias if we compute traditional estimators : the mortality rate q_j, within a time interval $I_j = [\tau_j, \tau_j + 1]$, can no longer be estimated with the ratio of the deaths, d_j, on the number of alive subjects at the beginning of the interval, l_j :

$$q_j = \frac{d_j}{l_j} = \frac{l_j - l_{j+1}}{l_j}$$

The quantity $\frac{d_j}{l_j}$ is indeed an inaccurate estimation as deaths that occur after censorship, but still *within* I_j (before τ_{j+1}), will not be known. For example, if an insured terminates his life insurance policy in February, and dies 4 months later that same year in June, his death would remain unknown to the life insurer who would therefore not take him into account among the d_j deaths in his portfolio for that year, whereas it is still a death at year I_j.

To compensate for the withdrawal, the number of alive subjects, l_j, is replaced by the number of subjects *exposed to risk*, meaning that subjects censored in the course of that year are *weighted* based on their length of presence in the portfolio. Depending on the hypothesis made on mortality, several types of exposure can be considered.

Three types of exposures are considered by actuaries: *Distributed exposure*, *Initial exposure* and *Central exposure*. However, we will only focus on the last two as the distributed exposure method relying on uniform distribution of deaths is not widely-used.

2.2.1 Initial Exposure and Balducci hypothesis

We denote **initial exposure** the quantity, EI_j, which represents the global amount of time each life was exposed to the risk of death during the interval j. As the exposure is based on the lives at the start of the interval the exposure can be referred to as initial. EI_j is the aggregation of the following individual exposure, ei_j:

- Alive at the start and the end of the interval are assigned 1
- Deaths during the time interval are assigned 1
- Censored are assigned the fraction of the interval they were observed

Formally, if we denote respectively $c_{i,j}$ and $t_{i,j}$ the censoring and death time of the individual i in interval j, w_j the number of withdrawals and l_j the number of alive subjects, the **initial exposure** is expressed as:

$$\begin{aligned} EI_j &= \sum_{i}^{l_j} 1_{\{t_{i,j}>1\}} \times 1_{\{c_{i,j}>1\}} + 1_{\{t_{i,j}<1\}} + c_{i,j} 1_{\{c_{i,j}<1\}} \\ &= \sum_{i}^{l_j} 1 - 1_{\{c_{i,j}<1\}} + c_{i,j} 1_{\{c_{i,j}<1\}} \\ &= l_j - w_j + \sum_{i=1}^{w_j} c_{i,j} \end{aligned}$$

To understand the idea behind this quantity, we define the two following notations for the rate of mortality,

- $q_j = P(T \leq \tau_j + 1 | T > \tau_j)$ in interval j
- ${}_{c_{i,j}}q_j = P(T \leq \tau_j + c_{i,j} | T > \tau_j)$ for the one in the interval $[\tau_j, c_{i,j}]$

The number of deaths can be expressed as the sum of the deaths observed within the interval and the deaths expected for the censored subjects. Formally:

$$d_j = (l_j - w_j)q_j + \sum_{i=1}^{w_j} {}_{c_{i,j}}q_j = l_j q_j - \sum_{i=1}^{w_j} {}_{1-c_{i,j}}q_{j+c_{i,j}} \tag{2.4}$$

The **Balducci hypothesis** supposes that mortality rates decrease over the interval and are defined as:

$$\begin{aligned} {}_{1-c_{i,j}}q_{j+c_{i,j}} &= P(T_i \leq \tau_j + 1 | T_i > \tau_j + c_{i,j}) \\ &= (1 - c_{i,j})P(T_i \leq \tau_j + 1 | T_i > \tau_j) = (1 - c_{i,j})q_j \end{aligned}$$

Injecting it in the previous equation gives:

$$d_j = l_j q_j - q_j \sum_{i}^{M} (1 - c_{i,j})$$

Solving the formula for q_j:

$$\hat{q}_j = \frac{d_j}{l_j - \sum_{i=1}^{w_j}(1 - c_{i,j})} = \frac{d_j}{l_j - w_j + \sum_{i=1}^{w_j} c_{i,j}} = \frac{d_j}{EI_j}$$

We finally get the rate of mortality estimator corrected for censoring with the previous definition of **initial exposure** as expected. This approach relies on *Balducci assumption*, which generally does not fit well for mortality as mortality rates increase with time. However withdrawals are usually small compared to the population, which allows to ignore these errors.

2.2.2 Central Exposure and Constant hazard function

Depending on the mortality observed within a dataset, one may prefer to use another assumption: the constant hazard function over each time interval. In this case, another exposure should be used.

The **central exposure**, EC_j is the amount of time individuals are observed within the interval. The difference with the **initial exposure** is that only individuals who survived the whole time interval are assigned 1.

The *constant hazard function* assumption implies that the hazard is constant over each time interval. For $e \in [0,1]$, we denote h_j the hazard rate over the interval $[\tau_j, \tau_j + 1]$:

$$h(\tau_j + e) = h_j \tag{2.5}$$

As long as we consider time interval small enough this hypothesis is acceptable.
When h_j is known for each j, the survival function is easy to compute :

$$S(\tau_j + e) = \exp(-\int_0^{\tau_j+e} h(s)ds) = \exp(-\sum_{s=1}^{j-1} h_s + eh_j) \tag{2.6}$$

The goal is then to estimate each h_j.

Let $ec_{i,j}$ be the **individual central exposure**, it corresponds to the amount of time one is observed within an interval. In addition, $\delta_{i,j}$ is a death indicator in $[\tau_j, \tau_{j+1}]$ (1 if death is observed, 0 otherwise). The likelihood can then be written as:

$$L = \prod_i S(\tau_j + ec_{i,j})h(\tau_j + ec_{i,j})^{\delta_{i,j}} \tag{2.7}$$

Using the constant hazard function assumption and considering the logarithm of the likelihood we get:

$$\log(L) = \sum_i [ec_{i,j}h_j + \delta_{i,j}\log(h_j) - \sum_{s=1}^{j-1} h_s] \tag{2.8}$$

The maximum likelihood estimator $\hat{h}_j$, so that $\frac{d}{dh_j}log(L) = 0$, is then the ratio of the number of death observed within the interval divided by the exposure :

$$\hat{h}_j = \frac{\sum_i \delta_{i,j}}{\sum_i ec_{i,j}} = \frac{d_j}{EC_j} \tag{2.9}$$

By definition, we can write $\hat{q}_j = 1 - exp(-\hat{h}_j)$. As initial exposure, the central exposure is interesting as it can be expressed through a closed formula. However, it relies as well on a death distribution, which is generally not verified in practice. Indeed, constant hazard rates could be considered as an accurate approximation for younger people, where a large proportion of deaths is accidental or due to critical illnesses (cancer, cardiovascular...), but it becomes far less accurate reaching very old ages : for example, the winter flu epidemic is responsible for a large number of elderly people deaths each year, which would encourage U-shaped hazard rates on calendar year intervals. Reaching even greater ages (99+) where the life expectancy becomes very short, every month sees a notable increase of the hazard rate, favoring increasing hazard rate functions over intervals.

2.2.3 Modeling our data using exposure

The main advantage of discretization is that it allows considering classical modeling approaches, by predicting the number of deaths for each time interval. In practice, we will model the random variable d_j describing the number of deaths using the exposure as weights or offset. Our database has been built on discrete annual time steps : each insured's biomedical features are measured on January 1^{st} every year, and each begun year of the contract is considered as due to the insurer, in other words, the contract is not terminated until December 31, the whole year's premium is due and the insured remains covered against death until the end of that year. Furthermore, death or censorship are only observed on December 31^{st}. It is basically an initial exposure setup where the censorship-induced exposure is always 1. This simplifying hypothesis allows us to have an annual exposure that is always 1.

2.3 Mortality modeling

2.3.1 Non-parametric estimators

Mortality tables

In actuarial science, mortality tables show, for each age and gender, the probability that a person will die before their next birthday ("probability of death"). They are widely used in life insurance to forecast annual death rates, probability of surviving any particular year of age, remaining life expectancy, or longevity risk. Typically, they report, for each age and gender, the number of observed deaths per 100,000 individuals in that group on the reference year.

There are two types of life tables used in actuarial science. The period life table represents mortality rates during a specific time period of a certain population. A cohort life table, often referred to as a generation life table, is used to represent the overall mortality rates of a certain population's entire lifetime : they must have had to be born during the same specific time interval. A cohort life table is more frequently used because it is able to make a prediction of any expected changes in mortality rates of a population in the future, and analyzes patterns in mortality rates that can be observed over time. Both of these types of life tables are created based on an actual population from the present, as well as an educated prediction of the experience of a population in the near future. Other life tables in historical demography may be based on historical records, although these often understate infant mortality in regions where statistical records of births are imperfect.

All mortality tables are specific to environmental and life circumstances, and are used to probabilistically determine expected maximum age within those environmental conditions. For example, discovery of a breakthrough cancer treatment or a global pandemic are events that modify the medical context under which a life table was built.

For example, my data generation process relies on France's 2016 mortality table : it reports, for each age and gender, the number of observed deaths per 100,000 in 2016 (see annex 5.5).

Kaplan-Meier Estimator

When we have no censored observations in the data, the empirical survival function is estimated by:

$$\hat{S}(t) = \frac{1}{n} \sum_{i=1}^{n} 1_{t_i > t}$$

This estimation is no longer viable in presence of censor as we do not observe the death time t_i but the end of observation time y_i. The non-parametric estimation is extended to censored data.

Kaplan-Meier estimator is the most widely used because of its simplicity to compute. It is implemented in many of survival libraries and packages of statistical and mathematical software. Besides, this estimator relies on no assumption and can thus easily be used as a reference model or to test hypothesis.

The main idea behind this estimator is that surviving after a given time t means being alive just before t and do not die at the given time t. Consequently, with $t_0 < t_1 < t$ we get :

$$\begin{aligned} S(t) &= \mathbb{P}(T > t) \\ &= \mathbb{P}(T > t_1\ ,\ T > t) \\ &= \mathbb{P}(T > t \mid T > t_1) \times \mathbb{P}(T > t_1) \\ &= \mathbb{P}(T > t \mid T > t_1) \times \mathbb{P}(T > t_1 \mid T > t_0) \times \mathbb{P}(T > t_0) \end{aligned}$$

In the end, by considering all the distinct times $t_i, (i = 1, ..., n)$ where an event occurred ranked by increasing order (whatever it is a death or censorship) we get:

$S(t_j) = \mathbb{P}(T > t_j) = \prod_{i=1}^{j} \mathbb{P}(T > t_i \mid T > t_{i+1})$, with $t_0 = 0$.

Considering the following:

d_j the number of deaths that occurred in t_j

N_j the number of individuals alive just before t_j

The probability $q_j = \mathbb{P}(T \leq t_j \mid T > t_{j-1})$ of dying in the time interval $]t_{j-1}, t_j]$ knowing the individual was alive in t_{j-1} can be assessed by : $\hat{q}_j = \frac{d_j}{N_j}$

Let δ_i be the censorship indicator of each observation; the Kaplan-Meier estimator is then defined as:

$$\hat{S}(t) = \prod_{t_i \leq t} (1 - \frac{d_i}{N_i})^{\delta_i} \tag{2.10}$$

We finally obtain a step function for the survival function where the jumps are observed at the empirical observed death times.

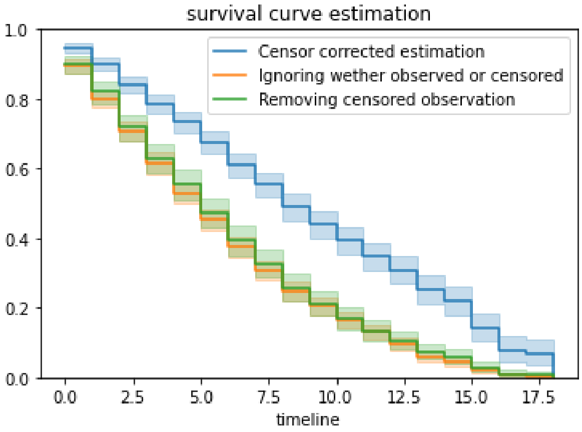


Figure 2.4: Impact of ignoring censoring in life duration study

As introduced before, ignoring censoring leads to an underestimation of the life duration. The Figure 2.4 highlights this underestimation. Three 'Kaplan-Meier' survival curves are plotted on different datasets: the real one, the one relying only on the fully observed individuals, and the one, for which censor and dead individuals are not distinguished. As the two last curves are below the real one, it means that at each time the survival probability is lower and thus that the risks have been overestimated.

Kaplan-Meier estimation is effective to get the survival curve of the global population. However, the precision of the estimation relies on the number of observations. If we want to take into account individuals' characteristics, we need to recompute the estimator for each chosen subset, which reduces the number of observations and thus the accuracy.

On the business side, it is indeed important to have a good prediction among different subgroups rather than on the global level. The insurer portfolio may indeed have an over-representation of some individuals compared to the population used to build the model, knowing that the insured population has lower mortality compared to the global population.

Nelson-Aalen Estimator

Instead of estimating the survival function, another method has been developed to estimate the cumulative hazard function. It is defined as:

$$\hat{H}(t) = \sum_{t_i \leq t} \frac{d_i}{N_i} \delta_i$$

To get an estimator of the survival function, one only has to plug-in the cumulative hazard estimator into the formula $S(t) = e^{-H(t)}$

$$\hat{S}(t) = e^{-\hat{H}(t)} = \prod_{t_i \leq t} (e^{-\frac{d_i}{N_i}})^{\delta_i} \approx \prod_{t_i \leq t} (1 - \frac{d_i}{N_i})^{\delta_i}$$

If the number of deaths is small compared to the number of people at risks at each time, the Nelson-Aalen plug-in survival function can be approximated by the Kaplan-Meier estimator. The two estimators are thus numerically close, but they have different properties, which implies different confidence intervals or median times.

2.3.2 Parametric extimators

Logistic Regression

In statistics, the logistic model (or logit model) is used to model the probability of a certain class or event $y_i \in \{0, 1\}$, such as pass/fail, win/lose, alive/dead or healthy/sick. In its basic form it uses a logistic function to model a binary dependent variable (in our case, survival at the time step).

The logistic function is of the form

$$p(x) = (1 + e^{-\lambda(x-\mu)})^{-1}$$

where x is the vector of covariates, $p(x)$ the estimated probability of survival, and λ and μ the parameters to optimize.

The likelihood function is

$$L = \prod_{y_i=1} p_i \prod_{y_i=0} (1 - p_i)$$

From that we can deduce the log-likelihood function, widely known as the binary cross-entropy loss, based on which we estimate the parameters :

$$l = \sum_i y_i ln(p_i) + (1 - y_i) ln(1 - p_i)$$

Poisson regression

Poisson regression model assumes that the total number of deaths within the time interval j follows a Poisson distribution and is mainly based on the **central exposure**. That is to say:

$$d_j | X \sim \mathscr{P}(EC_j h_j(X)) \tag{2.11}$$

The idea behind the parameter used in the Poisson distribution comes from the constant hazard function hypothesis, as under this hypothesis $\hat{h}_j = \frac{d_j}{EC_j}$, so that the expected values match.

In literature, d_j corresponds to an aggregate number of deaths for all similar individuals (i.e. with the same vector X of characteristics for a specific interval). However due to the additive property of the Poisson distribution, it is equivalent to consider afterward the aggregation of the prediction of the death indicator of everyone, which means considering a model as follows:

$$\delta_{i,j} | X \sim \mathscr{P}(ec_{i,j} h_{i,j}) \tag{2.12}$$

Using a log-link function, the model becomes equivalent to a classical Poisson regression model with the exposure in offset:

$$\log(E[d_j | X]) = \log(EC_j) + X'\beta = \log(EC_j) + \log(h_j) \tag{2.13}$$

which means

$$\log(h_j) = \log(\frac{E[d_j | X]}{EC_j}) = X'\beta \tag{2.14}$$

We then apply the classical generalized linear model with a Poisson distribution to a pseudo data table with exposure. Through the likelihood optimization with respect to β, we get the risk parameters:

$$L(\beta | X, D, E) = \prod_j \frac{(EC_j e^{X'_j \beta})^{d_j} e^{-EC_j e^{X'_j \beta}}}{d_j!} \tag{2.15}$$

It is also possible to consider an extension and add a regularization factor to only consider the variables with a high explanatory power. When the probabilities are small if the *central exposure* is not available in the data, approximating the model with *initial exposure* predicts similar results.

Using the maximum likelihood, the the hazard function estimator $\hat{h}_j = \exp(X'\hat{\beta}_j)$ gives us the mortality rate :

$$q_j = 1 - \exp(-\hat{h}_j) = 1 - \exp(-\exp(X'\hat{\beta}_j))$$

Cox Proportional Hazard Model

Cox's model allows to take into account the effect of covariates and to measure their impact on survival probabilities through the estimation of the hazard function. It becomes then possible to rank people's risk according to their physiological, familial, sociological or medical characteristics. This model may be considered as a regression model for survival data.

The standard proportional hazards Cox model relies on the fundamental hypothesis that the hazard rate function can be split into the product of two independent functions, one called the non-parametric baseline hazard function relying only on time, and the other called the relative risk function relying only on the different covariates.

$$\mu(t|Z) = \mu_0(t) \times exp(g_\beta(Z)) \tag{2.16}$$

Z the vector of covariates, which must be time-independent ;

g a function, linear or not, parameterized by β, that characterizes the impact of each individual's covariates on the hazard rate function independently of time ;

μ_0 the baseline hazard function : it corresponds to the function giving, at each timestep t, the hazard function for individuals with $Z = 0$; it is the time component in the hazard rate.

In the linear case, we have $g_\beta(Z) = \beta' Z$; we have not included a constant, since the exponential makes it equivalent to simply multiplying the baseline hazard function by a factor equal to the exponential of that constant. Specifically, the Cox proportional hazards model assumes a multiplicative effect of the covariates:

$$\mu(t|Z) = h_0(t) exp(< \beta, Z)$$

The phrase *proportional hazards* comes from the scalar product $< \beta, x >$ in the exponential that can be decomposed (supposing a covariates vector x of size n):

$$exp(<\beta, x>) = exp(\sum_{1}^{n} \beta_i x_i) = \prod_{1}^{n} exp(\beta_i x_i)$$

The effects of each individual feature of the covariates vector are multiplicative, "proportional". Given two individuals A and B, the ratio of their hazard functions is assumed to be unchanged over time, constant at each time step and depending only on the parameters β and their covariates Z_A, Z_B.

In the case of Cox neural networks, the function g is modeled by a neural network.

The main interest of the model is the possibility to rank people on their risk level without computing the survival function. The relative risk is introduced to this end:

$$RR = \frac{\mu(t|Z_A)}{\mu(t|Z_B)} = exp(g(Z_A) - g(Z_B)) \tag{2.17}$$

The estimation can be divided into two steps. Depending on the purpose of the study one may stop at the first one.

1. We compute the estimator $\hat{\beta}$ by maximizing the partial likelihood function defined by Cox :

$$L(\beta) = \prod_{i=1}^{m} \frac{e^{Z'_{j(i)}\beta}}{\sum_{j \in R_i} e^{Z'_j \beta}} \tag{2.18}$$

m the total of uncensored individuals

$j_{(i)}$ the individuals who died at time $t_{(i)}$

$t_{(i)}, ..., t_{(m)}$the ordered time of observed death events

R_i the risk set, which is a set of indices of the subjects that still alive just before $t_{(i)} : R_i = \{j : t_j \leq t_{(i)}\}$

When one is only interested in comparing the survival curve to classify individuals according to their survival probabilities, only the estimation of the risk parameter β is needed. The baseline hazard $\mu_0(t)$ does not only have any effect on the relative risk.

2. If we want the survival function for all individuals, the baseline hazard $\mu_0(t)$ is required in addition of the β parameters. The survival function can be computed as follows:

$$\hat{S}(t) = exp(-H_0(t)exp(Z\hat{\beta})) = S_0(t)^{exp(t\hat{\beta})} \tag{2.19}$$

where $H_0(t)$ is the cumulative baseline hazard function, and $S_0(t) = exp(-H_0(t))$ represents the baseline survival function. The Breslow's estimator is the most widely used method to estimate $\hat{S}_0(t) = exp(-\hat{H}_0(t))$ where

$$\hat{H}_0(t) = \sum_{t_i < t} \hat{h}_0(t_i) \text{ with } \hat{h}_0(t_i) = \frac{1}{\sum_{j \in R_i} e^{Z_j \beta}} \text{ if } t_i \text{ is a time event, 0 otherwise}$$

Estimating $S_0(t)$ thanks to Kaplan-Meier implies similar results.

In practice, the separation of the baseline hazard function modeling the time component of the hazard rate from the relative risk function modeling the covariates effect is subject to discussion. Indeed, the impact of some medical indicators is not the same through time : for example, a "healthy BMI" does not cover the same range of values for children (weight-height relation varies rapidly due to growth spurts), middle-aged or elderly (who experience a muscle loss) individuals.

Markov chains and semi-Markov models

Within the statistical toolbox for survival analysis, Kaplan-Meier (see 2.3.1) estimation and Cox proportional hazards regression (see 2.3.2) are commonly employed methods but not appropriate for all studies, particularly in the presence of competing risks and when multiple or recurrent outcomes are of interest. In that regard, Markov chain models can accommodate censored data, competing risks (informative censoring), multiple or recurrent outcomes, frailty[1], and non-constant survival probabilities.

A Markov model is a stochastic model describing a sequence of possible events (or a sequence of moves across possible states) in which *the probability of the next event depends only on the current state.* There is a finite number of defined states, one (and only one) of which must contain the individual at any particular time. These possible states may be, for example, {alive, dead}, {sick, healthy, dead}, {autonomous, dependent, dead}, etc. Markov chain models are useful for sequential data, but require caution since they are no-memory : for example, in cancer survival studies, it is often crucial to know whether a healthy policy-holder has always been healthy or is in remission of a past cancer (his chances of relapse are therefore much higher than the chances of another comparable healthy person of developing cancer).

Markov chain models allow to calculate the probability and rate (or intensity) of movement associated with each transition between states within a single observation cycle as well as the approximate number of cycles spent in a particular state. When observations are made at regular intervals, the number of cycles can be interpreted as time in a state. Time spent in all states prior to absorption can be summed to estimate the

---

[1] Frailty is an unobserved random proportionality factor that modifies the hazard function.

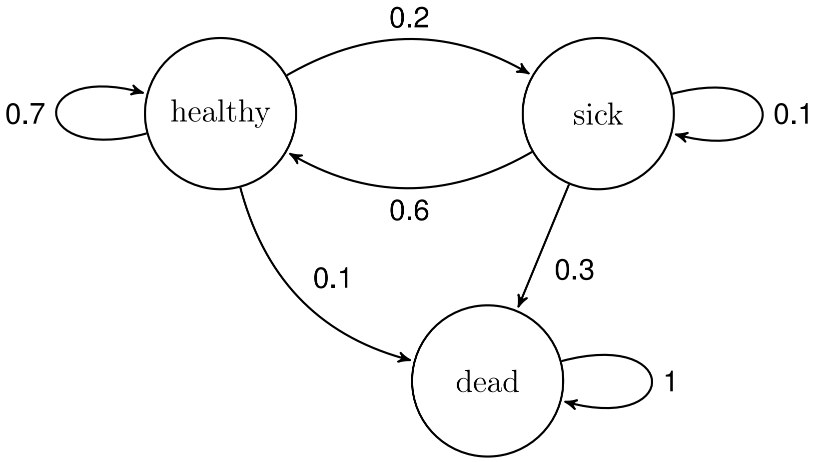


Figure 2.5: An example Markov model for survival analysis

total survival time. They also enable the estimation of survival times in multiple states: this is particularly attractive for studies of chronic diseases with well-defined phases, like cancer and autoimmune diseases, where remission and recurrence are of interest in addition to overall survival.

One only measures the state reached at every period beginning : "internal" transitions taking place within the time-lapse of a period are not taken into account. This is generally not a very restrictive hypothesis, since in survival analysis the studied events are not supposed to be very frequent (for example, it is rather unlikely that several transitions from *active cancer* to *remission* would happen within a time interval of one year).

Markov chains are particularly adapted to censored data : every individual's transition chain contributes information to the model, whether partial (the individual never reaches an absorbing state due to censorship) or complete (several transitions occur until an absorbing state is reached). Markov models also enable left truncation because individuals are not required to enter the transition matrix in any particular state. Possible movements across states, either uni- (e.g. autonomous → dependent) or bi- (e.g. sick ↔ healthy) directional, are depicted with a transition matrix or state diagram (see figure 2.5). In order for the process to terminate, at least one of the states must be absorbing, i.e., individuals have zero probability of leaving: death, for example.

Use of Markov chains requires two fundamental assumptions :

- Transition probabilities are constant over time (time homogeneity) ;
- The probability of the next transition depends only on the current state (first-order Markov property).

These two hypotheses are very strong to make in the general case :

- In survival analysis, transition probabilities may very often be dependent on a time factor, thus contradicting the time homogeneity hypothesis (for example, the probability of the transition autonomous → dependent is probably much higher at higher ages than in the youth). The time homogeneity assumption is often difficult to meet, particularly in studies of chronic disease where studies are years long, single observation cycles can span a year or more and increasing age generally corresponds to greater risk of disease or death. However, this concern can be mitigated by data stratification (e.g., by age group) or regression modeling, where the effect of covariates is included in the estimation of transition probabilities.
- The transition probability from one state to another is very rarely blind to the process anteriority, thus contradicting the first-order Markov property (for example, in the case of cancer relapse, the probability of transitioning *healthy* → *sick* is often much higher if there has already been *healthy* → *sick* → *healthy* transitions in the past than if the individual has always been stuck at *healthy* before.

In closing, given that improper use of Markov models may result in biased estimation, we will not focus on them in the context of this thesis since the two fundamental underlying hypotheses are clearly not verified.

2.4 Machine learning models in survival analysis

The introduction of machine learning in survival analysis has posed numerous challenges. It brings powerful predictive power, but also a "black box" problem in so far as some models can be hard to interpret and it can sometimes be complicated to justify the adequacy of the results. This can be a challenge, for example, to justify of sound pricing with respect to the risk in front of the regulator, or to guarantee the absence of discriminatory rules taking place inside the black box process. However, some interpretable models like Generalized Linear or Additive Models (GLMs, GAMs) work well on medium-sized amount of data.

Another challenge can be posed by the assumption of independence between the covariates that is underlying to most machine learning models. This is generally not true and can lead to big performance and reliability issues, as will be illustrated in the results section. For example, it couldn't be more wrong to assume independence of systolic and diastolic blood pressures, or even of blood pressure levels and BMI ; you can refer to the correlations analysis in the data generation par (figure 3.12).

2.4.1 Cox-Model adaptations

Several Machine Learning methods have been adapted to Cox's Proportional-Hazard models such as Trees, Neural-Networks, Generalized Additive Models, etc. In this section, we present Elastic Net and Gradient Boosting Machine adaptation, as they are the most widely used in practice.

Cox-ElasticNet

In *Regularization Paths for Cox's Proportional Hazards Model via Coordinate Descent* (2018), Tibshirani et al proposed to apply the Elastic Net regularization to Cox proportional hazard model. In statistics and, in particular, in the fitting of linear or logistic regression models, the elastic net is a regularized regression method that linearly combines the L1 (norm-1 of the parameters) and L2 (euclidean norm of the parameters) penalties of the lasso and ridge methods. The Cox-Elasticnet extends this approach to the Cox regression. The goal is to put aside the less relevant features by penalizing the models with a high number of parameters. Decreasing the number of features allows to diminish the signal noise and consequently increase model accuracy.

This approach is very useful in high dimensions, i.e. when the number of features is close to the number of observations. This might occur during some epidemiological studies where a significant amount of information is available for each patient, but a limited number of patients are observed.

In practice, this approach is often used to quickly identify variables that have the biggest explanatory power and to put aside the non-relevant ones. In this approach the model parameters β are estimated by optimizing the following:

$$\hat{\beta} = \underset{\beta}{\text{argmax}} \left[\log(L(\beta)) - \lambda \left(\alpha \, ||\beta||_1 + (1 - \alpha) \frac{1}{2} \, ||\beta||_2^2 \right) \right] \tag{2.20}$$

with L the likelihood of the Cox model (cf equation 2.18), λ and α hyper-parameters.

The penalization intensity is controlled thanks to the hyper-parameter λ. If λ equals 0 then we are performing a classical Cox regression. The higher the value of λ the higher the penalization, the lower the number of non-null parameters.

When the hyper-parameter α equals 0 it is called LASSO[2] regression, when α equals 1 it is named a Ridge regression. Hyper-parameter α is in $[0, 1]$, it balances between LASSO and Ridge regression.

This approach has the same issues as the classical in Cox model as it relies heavily

---

[2] LASSO stands for Least Absolute Shrinkage and Selection Operator.

on the proportional force of mortality assumption that might not be verified. Besides, non-linear effects and unspecified interactions will not be captured by this model.

Cox - Gradient Boosting Machine

A gradient boosting adaptation can help integrating non-linear effects within the Cox framework.

Gradient boosting consists in aggregating several simple models (*weak learners*). The weak learners remain the same base learners throughout the process, but are iteratively trained on the residual errors made by their predecessor. Thus, each model relies on those constructed in previous step.

Gradient Boosting Machine is a mix between gradient boosting and gradient descent, which is an optimisation process to minimize a loss function. The adaptation of the GBM to a Cox's proportional hazard model is possible by choosing the opposite of Cox partial likelihood as loss function :

$$LL(\beta) = -\sum_{i=1}^{m}[X'_{j(i)}\beta - log(\sum_{j \in R_i} e^{X'_{j(i)}\beta})] \tag{2.21}$$

Generally speaking, the algorithm is presented as the process below:

Initialisation : $F_0(x) = argmin_{\beta} LL(\beta)$

For $m = 1$ to M *(number of weak learners):*

- Computation of the pseudo-residuals: $r_m = -\frac{dL(F_{m-1}(X))}{F_{m-1}(X)}$
- Fitting a new weak learner on pseudo-residuals: $f_m(X) = r_m$
- Finding the best γ_m by solving $\gamma_m = argmin_{\gamma} L(F_{m-1}(X) + \gamma \times f_m(X))$
- Update the new model: $F_m = F_{m-1} + v \times \gamma_m f_m$

Thus, at each iteration, until the stopping condition is satisfied, we try to reduce the global error by fitting each specificity of the residuals. A learning rate, γ_m is introduced to control how much we adjust the weights of our base learner. This parameter may be constant and chosen at the beginning of the process or optimized at each step. A large value may reduce computing time but may cause divergence, a small one ensures convergence and getting an optimum but make learning time more consuming. The shrinkage parameter v, a scalar between 0 and 1, allows to regularize the model and ensure the convergence of the model.

The real advantage of gradient boosting is that it can be adapted to any weak learners. Most of the time trees are chosen. Cox Gradient Boosting Machine is a way of building classical regression trees by taking into account censoring within the loss function and assuming the proportional hazard hypothesis. In this case, trees are constructed consecutively, and the gradient shows the best path so that each tree is constructed on the previous one in such a way as it leads to the biggest error reduction.

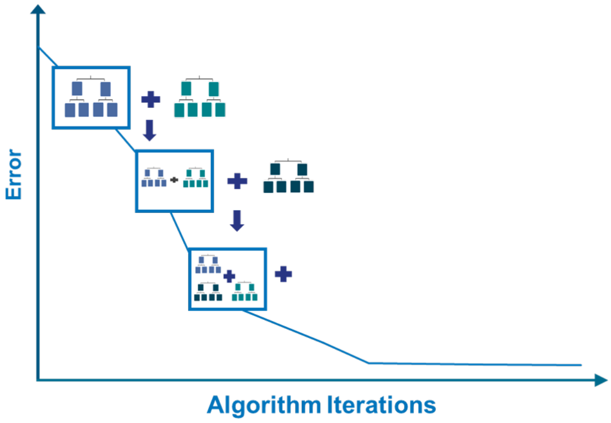


Figure 2.6: Gradient Tree Boosting

2.4.2 Survival Trees

Another method to build specific trees for survival analysis have been developed. Compared to Cox-Gradient Boosting, it enables to create predictor trees, which may be directly interpreted. The real advantage of trees is its simplicity compared to other Machine Learning techniques, which contributes to short computation time.

Survival trees are the direct adaption of decision trees to survival analysis. Traditional decision trees are also called CART (i.e. classification and regression tree), which is the fundamental algorithm of the tree-based methods family. The CART algorithm developed by Breiman makes the use of trees very popular to solve regression and multi-class classification problems.

Like CART, survival trees are binary trees grown by a recursive splitting of tree nodes. Starting from the root node composed of all the data, the observations are partitioned into two daughter nodes according to a predetermined criterion. Using a recursive process each node is split again into two nodes until reaching the *stopping criterion*. The best split for a node is found by searching over all possible variables and values, the one that maximizes survival difference.

The difference between *CART* and *Survival trees* relies on the **splitting criterion** used to build the tree. When dealing with survival data, the criterion must explicitly involve survival time and censoring information. Either it aims at maximizing the between-node heterogeneity or at minimizing the within-node homogeneity.

Log-rank criterion

The most widely used criterion is the maximization of the log-rank statistic (cf Annex 5.5) between the two sub-samples of the nodes, which contributes to creating splits that distinguish the most the mortality. As it is impossible to measure the similarity of the mortality within a group, the idea behind is that by sequentially creating splits with distinct mortality, we assume to obtain homogeneous groups at the end as the dissimilar cases become separated at each node.

Hyper-parameters should be introduced to optimize the number of splits: a minimum occurrence of events within a leaf or a lower threshold of the log-rank statistic to make a split. The intuition behind this *stopping criterion* is to ensure the quality of the split. The first one forces the splitting criterion to be computed on enough data to make sure that the log-rank statistic is consistent. The lower bound for the second one comes from the reject region bound of the underlying log-rank test, which means a node should not be split if the mortality is not statistically different with respect to any variable.

The main advantage of this method is that it does not rely on major assumptions to build the tree, even if the statistic considered to measure the difference in moralities between groups is questionable. Indeed, the log-rank test performance may be poor in some situations.

Once the tree is built, the model assumes that individuals within a leaf have the same common survival curve and thus a global survival curve is computed based on the individual within each final leaves. In open-source packages, the Nelson-Aalen estimator is used to compute the cumulative hazard function, from which we can deduce the survival curve or the expected lifetime duration. Experimental studies have shown that using the Kaplan-Meier estimator to directly estimate the survival curve gives similar results.

Thanks to the binary nature of survival trees, individuals with characteristics x_i fall into a unique leaf f composed of observations (x_i, δ_i) with $i \in \mathscr{I}_f$. The prediction of his cumulative hazard function is the estimator for x_i's terminal node:

$$\hat{H}(t|x_i) = \hat{H}_f(t) = \sum_{\substack{t_i < t \\ i \in I_f}} \frac{d_i}{N_i} \delta_i \tag{2.22}$$

Some other criteria have also been studied such as C-index maximization or deviance minimization within one node.

Deviance criterion

The deviance minimization is based on a likelihood estimation relying on the proportional hazard function to partitions the observation. Under this hypothesis, the hazard function within a leaf f composed of observations (x_i, δ_i) with $i \in \mathscr{I}_f$, is expressed as follows:

$$h_f(t) = h_0(t) \times \theta_f$$

Using the formula 2.3, the likelihood can thus be rewritten :

$$L = \prod_f \prod_{i \in I_f} h_f(t_i)^{\delta_i} S_f(t_i) = \prod_f \prod_{i \in I_f} h_f(t_i)^{\delta_i} e^{-H_f(t_i)} = \prod_f \prod_{i \in I_f} (h_0(t)\theta_f)^{\delta_i} e^{-H_0(t_i)\theta_f} \tag{2.23}$$

Where $H_0(t)$ and $h_0(t)$ are respectively the baseline cumulative hazard function and the baseline hazard function, and θ_f is the parameter to estimate by likelihood maximisation. When H_0 is known, we can get the maximum likelihood estimator:

$$\hat{\theta_f} = \frac{\sum_{i \in I_f} \delta_i}{\sum_{i \in I_f} H_0(t_i)}$$

In practice, the cumulative hazard function is unknown and we plug in the breslow estimator

$$\hat{H_0}(t) = \sum_{i:t_i \leq t} \frac{\delta_i}{\sum_f \sum_{i:t_i \geq t; i \in I_f} \theta_f}$$

The deviance is finally defined as:

$$R(f) = 2[L_f(saturated) - L_f(\hat{\theta_f})] \tag{2.24}$$

where $L_f(saturated)$ is the log-likelihood for the saturated model that allows one parameter for each observation and $L_f(\hat{\theta_f})$ is the maximal log-likelihood.

The tree-building algorithm is based on CART: it splits the observation and covariate space into regions that maximize the reduction of the deviance realized by the split by testing all possible splits for each of the covariates. In this approach a *stopping criterion* regarding the minimum size of a node is also considered since the likelihood estimation converges when it relies on a large amount of data.

Simulations have shown that the performance is similar to the log-rank statistic. However, this method is not assumption-free and may not be applied to all datasets.

Trees built with a C-index maximization also yield similar-quality results and are also assumption-free, but require much more computation time. Thus, trees using the log-rank criterion should be privileged.

Survival Random Forest

Random survival forest extends the random forest method to right-censored survival data.

Random forest is an ensemble method inspired by the *bagging* of decision trees. Bagging, which means *Bootstrap Aggregating*, is an ensemble learning method that enables to create more robust predictor thanks to the aggregation of several ones trained on different subsets.

Bagging consists in sampling with replacement B random subsets to train B trees $\hat{f}^b$ on these subsamples. Finally the prediction of a new input X is defined as:

$$\hat{f}_{Bagging}(X) = \frac{1}{B}\sum_{i=1}^{B}\hat{f}^b(X)$$

Random forests differ from a simple bagging of trees, as randomization is not only applied to drawn samples but also to select features. At certain nodes, rather than considering all the variables, a random subset of the attributes is selected to compute the splitting criterion. The introduction of randomization enables to reduce the correlation among the trees and to improve the predictive performance. The training process is illustrated in Figure 2.7.

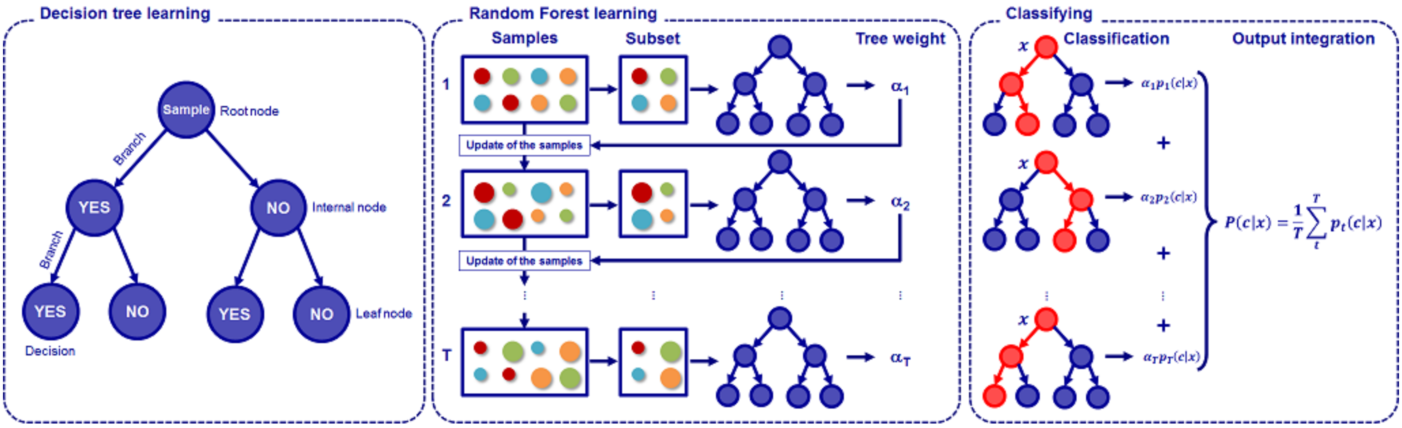


Figure 2.7: Random forest process illustration

Random survival forest is an ensemble tree method developed by Ishwaran et al that follows the same process but considers survival tree instead of traditional decision trees.

The algorithm is processed as below:

- Draw B bootstrap sample from the original data that excludes on average 37% of the data, called out-of-bag data (OOB data)
- Grow a survival tree for each bootstrap sample, by selecting at each node p candidate variables. The split is chosen among the candidate variable that maximizes the survival difference between leaves.
- Calculate the cumulative hazard function for each tree, $\hat{H}^b(t|x_i)$ and average it over all the trees to obtain the ensemble cumulative hazard function:

$$\hat{H}(t|x_i) = \frac{1}{B}\sum_{b=1}^{B}\hat{H}^b(t|x_i)$$

The interpretation of the result may be questionable as we average several hazard functions to get the predicted one. However as $H(t) = \int_0^t h(s)\mathrm{d}s$ is already a sum of functions, averaging it returns still a sum : $\hat{H}(t) = \frac{1}{B}\sum_{b=1}^{B}\int_0^t \hat{h}^b(s|x_i)\mathrm{d}s = \int_0^t [\frac{\sum_{b=1}^{B}\hat{h}^b(s|x_i)}{B}]\mathrm{d}s$ and the prediction makes sense.

The main advantage is that forests can model non-linear effects without any prior transformation of the data and contrary to boosting, in bagging each tree is built independently and the process can thus be paralleled.

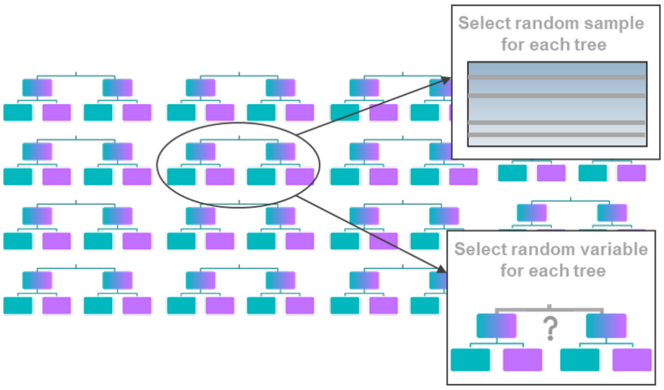


Figure 2.8: Bagging illustration

Chapter 3

Data Generation

3.1 Assumptions and key criteria

The observation of cohort events is not frequent in insurance for now. In this memoir, I simulate time series of biometric variables that are correlated with their mortality risk. In addition, I simplified the study framework with the following assumptions :

- Date of birth is 01/01/xxxx ;
- Date of death is 31/12/xxxx ;
- Censorship occurs between 01/01 and 31/12, and is observed on 31/12 ;
- Annual observation of biometric features (GP report) is issued on 01/01/xxxx

A consequent part of this study has been focused on generating realistic, real-world-like sequential data for a cohort of patients.

We focus on the impact of cardiovascular characteristics on the mortality of an insured. Also, the life duration is discrete and not continuous : once a year, the insured's medical parameters are measured and his survival or censorship reported.

The choice of probability laws and the parametrization of the initial distribution of the variables is based on the following references (see appendix 5.5) :

- 2016 mortality tables (by gender), France, INSEE ;
- 2016 gender ratios (by age), France, INSEE ;
- 2019 proportions of obese, hypertense, and smokers (by age category and gender), Canada, Statistique Canada.

Furthermore, based on the medical literature and a discussion with the doctors and underwriters at SCOR, certain variables are correlated :

- Corr(Obesity, Hypertension) : 0.40 ;

- Corr(Hypertension, Smoking status) : 0.40 .

3.2 Data generator schema

3.2.1 From an initial distribution...

Beta law

The beta distribution is relevant for the following reasons. The beta probability distribution has a closed support $[0,1]$ that is convenient for enforcing certain medical variables to take values in a range. Different values for the 2 shape parameters α and β can result in various forms of the beta distribution that reflect the variance or skewness of the observation.

We will designate by Γ the commonly used extension of the factorial function to strictly positive real numbers :

$$z > 0, \quad \Gamma(z) = \int_0^{+\infty} x^{z-1} e^{-x} dx$$

The beta distribution $\mathscr{B}(\alpha, \beta)$, parametrized by $\alpha > 0, \beta > 0$, has the following density function over $[0,1]$:

$$x \in [0,1], \quad f_{\alpha,\beta}(x) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

It is notable here that for $\alpha = \beta$, the law is symmetrical.

Its expectancy is $\mathbb{E}[X] = \frac{\alpha}{\alpha+\beta}$ and an estimator of its median is, for $\alpha > 1, \beta > 1$, $q_{0.5} \approx \frac{\alpha - 1/3}{\alpha+\beta-2/3}$.

Simulation of age and gender

We have simulated a population resembling a life insurer's portfolio. Therefore, the first step has been to simulate ages for $N = 200\_000$ insureds, following a beta law on [25, 74] :

$$X \sim \mathscr{B}(\alpha = 20, \beta = 20), \quad X_{age} = 25 + 49 \times X$$

From a theoretical point of view, the expectancy $\mathbb{E}[X_{age}] = 25 + 49 \times \frac{20}{20+20} = 25 + 24.5 = 49.5$ and the median $q_{0.5} \approx 25 + 49 \times \frac{20-1/3}{20+20-2/3} = 25 + 49 \times \frac{1}{2} = 49.5$ (it is logical to predict the same median and mean since the law is symmetrical).

After that, using the gender ratios by age :

$$(\text{number of men of age x})/(\text{number of women of age x})$$

provided in the INSEE table, we define, for each age x, the probability to be a man:

$$p_{man,x} = \frac{\text{number of men of age x}}{\text{number of people of age x}} = \frac{\text{gender ratio at age x}}{(\text{gender ratio at age x}) + 1}$$

For each generated individual of age x, we generate an is_M feature which is a gender indicator (equals 1 if the individual is a man) from drawing a Bernoulli random variable $\mathscr{B}(p_{man,x})$. Between ages 45 and 54, which include the majority (80.5%) of our portfolio, the gender ratio is of 0.99 ; therefore, we expect a gender ratio in our simulated database to be of the same order of magnitude. And indeed, with 49.4% of men at entry in the portfolio, we end up with a gender ratio of 0.98.

Multivariate normal distribution

I have used a multivariate normal distribution in order to introduce correlations in the initial distribution of some features. First, I generated $N = 200\_000$ samples from the following distribution:

$$\begin{pmatrix} is\_obese \\ is\_hypertense \\ is\_smoker \end{pmatrix} \sim N\left[\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & 0.4 & 0 \\ 0.4 & 1 & 0.4 \\ 0 & 0.4 & 1 \end{pmatrix}\right]$$

Then, on the 2019 Canadian data about obesity, hypertension and smoking habit preponderance (as a fraction of sub-population p) by age category and gender I applied Φ^{-1} the inverse of the cumulative distribution function for a gaussian random variable, giving me the corresponding quantiles $\Phi^{-1}(p) = q$ for a normal law. Finally, for each individual of given age and gender, I replaced the floating values x_i for the is_obese, $is\_hypertense$ and is_smoker by the indicator $\mathbb{1}(x_i < q)$ where q was the quantile for the normal law corresponding to the proportion p of the feature observed in the individual's age and gender sub-group.

From categorical to continuous features

Now that I am provided, for each individual, with an **age**, **gender** and **smoking status**, I would like to transpose the indicators or being obese and having hypertension into numerical data, that I will then make evolve through time, deriving risk factors from the levels and evolution. Once again, for the practical reasons explained before, I chose to simulate the BMI and blood pressure distributions across my entire insured population according to beta laws.

For the BMI, obesity is medically defined as a BMI superior to 30, and I decided, with advice from the biomedical risk team, to fix the *initial* attainable range of BMI values to $[15, 50]$. The obesity threshold in the $[0, 1]$ support of the beta law is located at $q_{obesity} = \frac{30-15}{50-15} = \frac{3}{7} \approx 0.4286$. I calculated the overall obesity prevalence in my population ($p_{obesity} = 37.5\%$).

My objective now is to determine the parameters (α, β) of a beta distribution that will ensure that $1 - F_{\alpha,\beta}(q_{obesity}) = p_{obesity}$, where $F_{\alpha,\beta}$ is the cumulative distribution function of the beta distribution. I fixed a parameter $\alpha = 2$ (completely arbitrary), the objective is now to solve for β :

$$\int_{q_{obesity}}^{1} x^{\alpha-1}(1-x)^{\beta-1} = p_{obesity}\frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

where α, $q_{obesity}$, $p_{obesity}$ are set. Alternatively we can solve for (α, β) with given $q_{obesity}$, $p_{obesity}$, but setting $\alpha = 2$ enables me to minimize the squared difference as a function of β.

A numerical minimization leads to the solution $\beta \approx 3.3438$.

The final step is to draw $N = 200\_000$ samplings from

$$X_{BMI} = 15 + 35 \times X, \quad X \sim \mathscr{B}(\alpha = 2, \beta = 3.3438)$$

and randomly assign the values $X_{BMI} \geq 30$ to obese individuals, the values $X_{BMI} < 30$ to non-obese individuals.

For the systolic and diastolic blood pressures, I proceeded in a similar manner. There are two types of medically recognised hypertension, "standard" high blood pressure defined by systolic blood pressure ≥ 140 and diastolic blood pressure ≥ 90, and isolated systolic hypertension where the diastolic blood pressure remains in the normal range of $[70, 90]$. Therefore, for the individuals classified as having hypertension, I randomly assigned 10% of them as having isolated systolic hypertension.

I then proceeded in a similar manner as with the BMI :

$$X_{systolic} = 200 - 100 \times X_1, \quad X_1 \sim \mathscr{B}(\alpha = 5, \beta = 1.6741)$$

$$X_{diastolic} = 200 - 130 \times X_2, \quad X_2 \sim \mathscr{B}(\alpha = 20, \beta = 2.0969)$$

The final step of assigning the numerical systolic and diastolic blood pressure measurement is different.

1. Sort the data in descending order based on *systolic* hypertension indicator, age category (10-year intervals), BMI category (5-point intervals), smoking status,

and finally $BMI \times age$;

2. Generate $N = 200\_000$ samples of $X_{systolic}$, sort them in descending order, assign them to the aligned ordered individuals ;
3. Sort the data in descending order based on *diastolic* hypertension indicator, age category (same as previous), BMI category (same as previous), smoking status, and finally $BMI \times age$;
4. Generate $N = 200\_000$ samples of $X_{diastolic}$, sort them in descending order, assign them to the aligned ordered individuals.

That procedure ensures that systolic and diastolic blood pressure are heavily correlated (97.5%), ensures almost surely that we have $systolic > diastolic$, respects the heavy impact of biometric factors such as age, BMI and smoking habit on blood pressure (figure **??**) without being "too" deterministic (ordering primarily by age and BMI *class* rather than directly on the granular values, and refining only as the last sorting key in order to avoid "jumps" in the average blood pressure values at the borders of age and BMI categories).

3.2.2 ...To a dynamic evolution...

Dynamic Features

Once we have an initial distribution of age, gender, smoking status, BMI and systolic and diastolic blood pressures for our $N = 200\_000$ individuals at entry in the portfolio (timestep 0), we want to generate dynamic evolution of these features. For simplification, I consider the smoking status to be static (smokers don't stop, non-smokers don't start). I also considered that I would assign trend signs for BMI and blood pressure (ascending, descending, stable) and that this trend isn't modified through time.

For the evolution of the body mass index and diastolic and systolic blood pressures, based on the medical literature and discussion with the biomedical risk team, I have chosen the trends as follows :

- Half of people have an increasing BMI, a quarter are stable and a quarter are decreasing ;
- Systolic and diastolic blood pressures always evolve in the same direction (either increasing, decreasing or stable) ;
- Half of people have an increasing blood pressure, 11.5% people have a decreasing blood pressure and 38.5% are stable ;

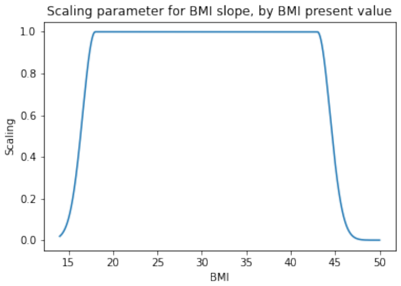


Figure 3.1: $f_{border}^{BMI}(X^{BMI})$

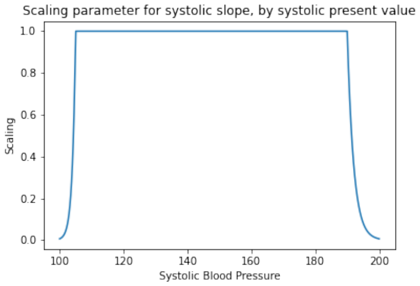


Figure 3.2: $f_{border}^{systolic}(X^{systolic})$

- There is a 0.2 correlation between BMI trend sign and blood pressure trend sign.

Note : you have to be careful here that these are only choices about the *trend signs*, increasing (+1), stable (0) or decreasing (-1), independently both from the amplitude of said evolution and from the initial values of these parameters.

At each time step (year) :

$$X_{t+1}^{BMI} = X_t^{BMI} + x_t^{BMI}, \quad x_t^{BMI} \sim \mathcal{N}(s^{BMI} \times \mu^{BMI} \times f_{border}^{BMI}(X_t^{BMI}), \sigma^{BMI})$$

where s^{BMI} is the BMI trend sign (0, +1 or -1), $\mu^{BMI} = 0.2$ is an arbitrary yearly average evolution of ±0.2 BMI points (equivalent to a win or loss of 0.6kg for a person measuring 1.73m), $\sigma^{BMI} = 0.3$ is an arbitrary standard deviation.

The final value X_{t+1}^{BMI} is finally clipped to the $[14, 50]$ range of values, and f_{border}^{BMI} is a function only used to decrease the evolution speed once the BMI values get nearer the edges of that interval, so that the BMI stabilizes more smoothly once it hits one of the clipping values (see figure 3.1).

I proceeded similarly with the blood pressure :
$\mu^{BP} = 1.5$, $\sigma^{BP} = 1$, $X_t^{systolic} \in [95, 200]$, $X_t^{diastolic} \in [65, X_t^{systolic} - 10]$, $f_{border}^{systolic}$ illustrated in figure 3.2 and $f_{border}^{diastolic}$ illustrated in figure 3.3.

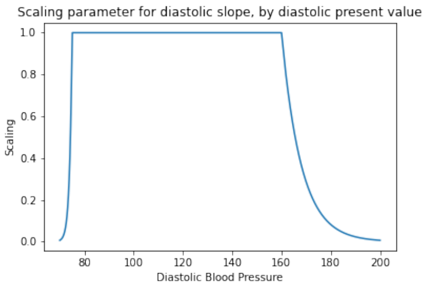


Figure 3.3: $f_{border}^{diastolic}(X^{diastolic})$

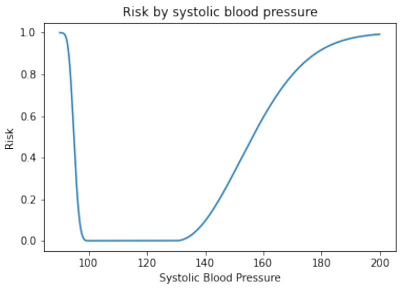


Figure 3.4: Systolic Risk Factor

3.2.3 ...Ending in Death or Censorship

For each individual and each time step t, I compute a risk score R_t :

$$R_t = 1 + \mathbb{1}_{smoker} + 2 \times (R_t^{BMI} + R_t^{BP})$$

The blood pressure risk factor R_t^{BP} is the sum of diastolic $R^{diastolic}(X_t^{diastolic})$ (figure 3.5) and systolic $R^{systolic}(X_t^{systolic})$ (figure 3.4) risk factors, each comprised in $[0,1]$, therefore $R_t^{BP} \in [0,2]$. The BMI risk factor R_t^{BMI} is the sum of 3 risk factors that depend on the instantaneous level of BMI (figure 3.6), its variability over the last 3 years (figure 3.7) and its evolution since the previous year (figure 3.8). Therefore, $R_t^{BMI} \in [0,3]$. Finally, we end up with a risk factor $R_t \in [1,12]$, characterizing, at given gender and age, the risk level of an individual with respect to his smoking, BMI and blood pressure characteristics, compared to the baseline individual, non-smoker, with BMI, systolic and diastolic blood pressure stable in the normal ranges. We multiply R_t by

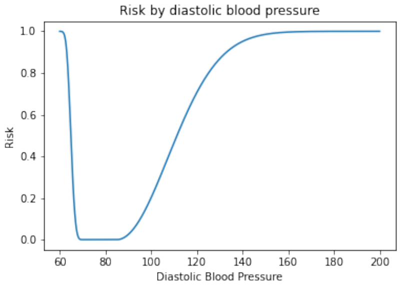


Figure 3.5: Diastolic Risk Factor

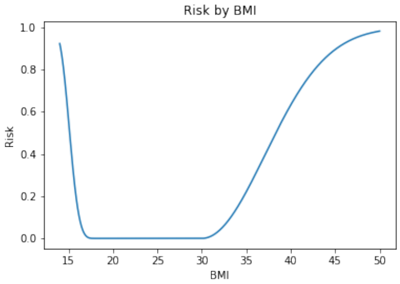


Figure 3.6: BMI Risk Factor

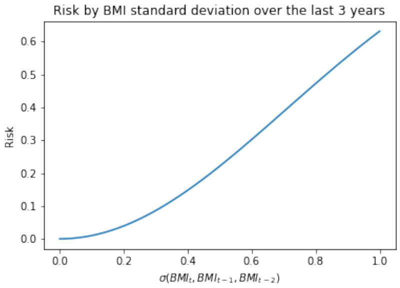


Figure 3.7: BMI Variability Risk Factor

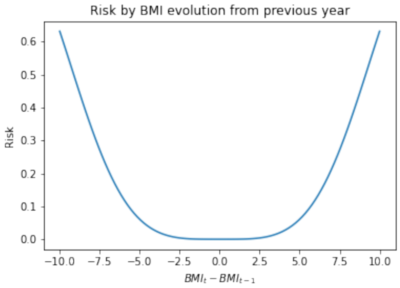


Figure 3.8: BMI Evolution Risk Factor

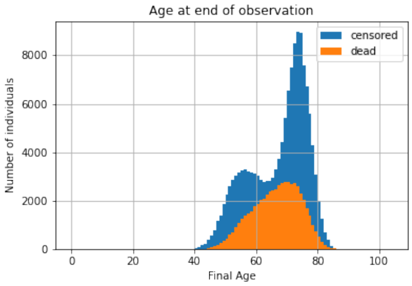


Figure 3.9: Distribution of age at final time step

the annual hazard rate for the gender and age $hr_{gender,age}$, and finally draw a Bernoulli random variable :

$$\mathbb{1}[\text{Survives year t}] \sim \mathscr{B}(min(0.99, R_t \times hr_{gender,age}))$$

That gives us an iterative generation of the lifespan T for all individuals.

Lastly, we generate a censorship variable C, independent from T and identically distributed across individuals : it is based on an annual lapse rate of 2% and a maximum observation history of 26 years, in line with the general specifics of a life insurance portfolio.

$$C = min(26, \hat{C}), \quad \hat{C} \sim \mathscr{N}\mathscr{B}(1, 0.02), \quad \hat{C} \perp\!\!\!\perp T$$

In the general setup I make the hypothesis that terminating a contract is only possible on the last day of each year, when we look up our individuals to check whether they are still alive, therefore if death and censorship were to happen in the course of the same year I would always consider my individual to be observed. Finally, we have :

$$Y = min(T, C), \qquad \delta = \mathbb{1}[T \leq C]$$

In my database I end up with 28.5% observed individuals. The end age distribution is represented in figure 3.9.

3.3 Descriptive statistics of the dataset

3.3.1 Initial distribution

Empirically, we obtain the following initial age distribution (see figure 3.10) : the minimal observed age is 34, the maximum 65, the average is at 49.5 and the median at 50. This corresponds rather well to the age of life insurance subscription observed by insurers on the commercial side.

We obtain the following initial BMI distribution across our initial population (figure

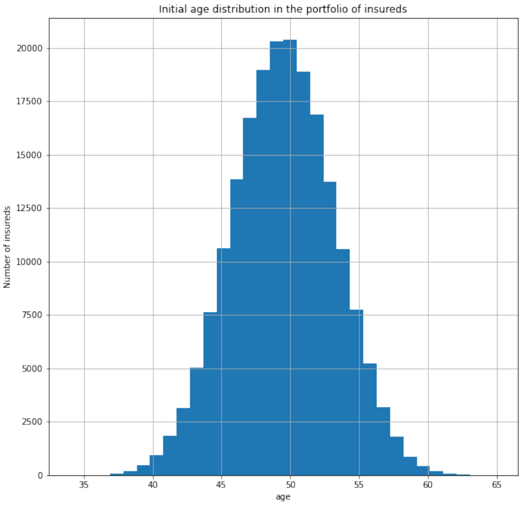


Figure 3.10: Initial age distribution in the portfolio

3.11), where you can notice an important limitation : although the position of BMI values under or above 30 is correlated to gender and age class based on the previous construction of indicators, the BMI level inside each class isn't. This is a deliberate choice, in order to avoid being too deterministic and to keep enough variability in the data.

Lastly, for the initial systolic and diastolic blood pressure value distribution, the order of magnitude and importance of the correlations observed is in line with what is reported in the medical literature and the expert opinion of the biometric risk modeling team (see table).

	age	is_M	smoker	bmi	syst.	diast.	risk	hr	Y	target
age	1.00	-0.00	0.02	-0.00	0.57	0.56	0.19	0.50	-0.10	0.20
is_M	-0.00	1.00	0.07	0.09	0.07	0.07	0.08	0.42	-0.09	0.18
smoker	0.02	0.07	1.00	0.01	0.24	0.23	0.68	0.42	-0.07	0.12
bmi	-0.00	0.09	0.01	1.00	0.43	0.43	0.35	0.25	-0.04	0.08
syst.	0.57	0.07	0.24	0.43	1.00	0.98	0.67	0.67	-0.13	0.23
diast.	0.56	0.07	0.23	0.43	0.98	1.00	0.70	0.69	-0.13	0.22
risk	0.19	0.08	0.68	0.35	0.67	0.70	1.00	0.76	-0.12	0.19
hr	0.50	0.42	0.42	0.25	0.67	0.69	0.76	1.00	-0.16	0.27
Y	-0.10	-0.09	-0.07	-0.04	-0.13	-0.13	-0.12	-0.16	1.00	-0.13
target	0.20	0.18	0.12	0.08	0.23	0.22	0.19	0.27	-0.13	1.00

In these correlations, $target$ refers to the indicator of whether the individual will be observed to die or will be censored. Of course, risky individuals are likelier to die early and therefore to have $target = 1$ than healthy individuals. Y is his length of

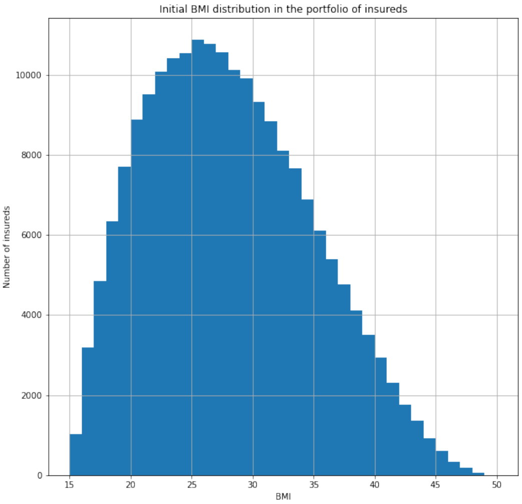


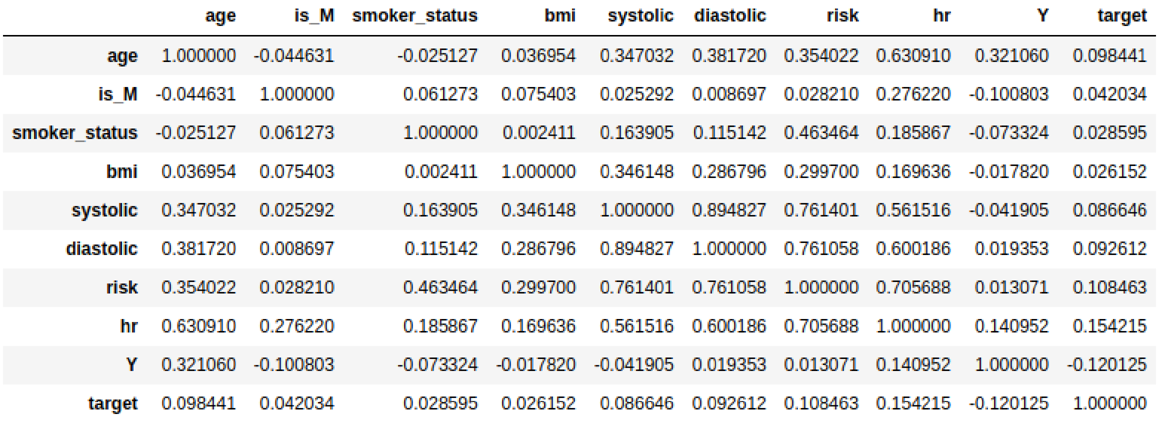
Figure 3.11: Initial BMI distribution in the portfolio

presence in the database, ending either with death or censorship: healthy individuals or women have more chances to stay long. *risk* is the individual's risk factor at entry in the portfolio, calculated on smoking habit, BMI and blood pressures, and *hr* is this risk score multiplied by the mortality tables hazard rates that depend solely on age and gender. However, we observe important correlations between *risk* and age or gender: this is due to the fact that, for example, more men smoke than women, or that hypertension becomes largely prevalent with aging.

3.3.2 Entire dataset

Correlations

The overall correlations between features (all timesteps) are explicited on figure 3.12.



	age	is_M	smoker_status	bmi	systolic	diastolic	risk	hr	Y	target
age	1.000000	-0.044631	-0.025127	0.036954	0.347032	0.381720	0.354022	0.630910	0.321060	0.098441
is_M	-0.044631	1.000000	0.061273	0.075403	0.025292	0.008697	0.028210	0.276220	-0.100803	0.042034
smoker_status	-0.025127	0.061273	1.000000	0.002411	0.163905	0.115142	0.463464	0.185867	-0.073324	0.028595
bmi	0.036954	0.075403	0.002411	1.000000	0.346148	0.286796	0.299700	0.169636	-0.017820	0.026152
systolic	0.347032	0.025292	0.163905	0.346148	1.000000	0.894827	0.761401	0.561516	-0.041905	0.086646
diastolic	0.381720	0.008697	0.115142	0.286796	0.894827	1.000000	0.761058	0.600186	0.019353	0.092612
risk	0.354022	0.028210	0.463464	0.299700	0.761401	0.761058	1.000000	0.705688	0.013071	0.108463
hr	0.630910	0.276220	0.185867	0.169636	0.561516	0.600186	0.705688	1.000000	0.140952	0.154215
Y	0.321060	-0.100803	-0.073324	-0.017820	-0.041905	0.019353	0.013071	0.140952	1.000000	-0.120125
target	0.098441	0.042034	0.028595	0.026152	0.086646	0.092612	0.108463	0.154215	-0.120125	1.000000

Figure 3.12: Correlations in the simulated database

Trends

Analyzing how BMI trends impact mortality highlights that early mortality is driven by extreme variations or dangerous weight values, see figure 3.13.

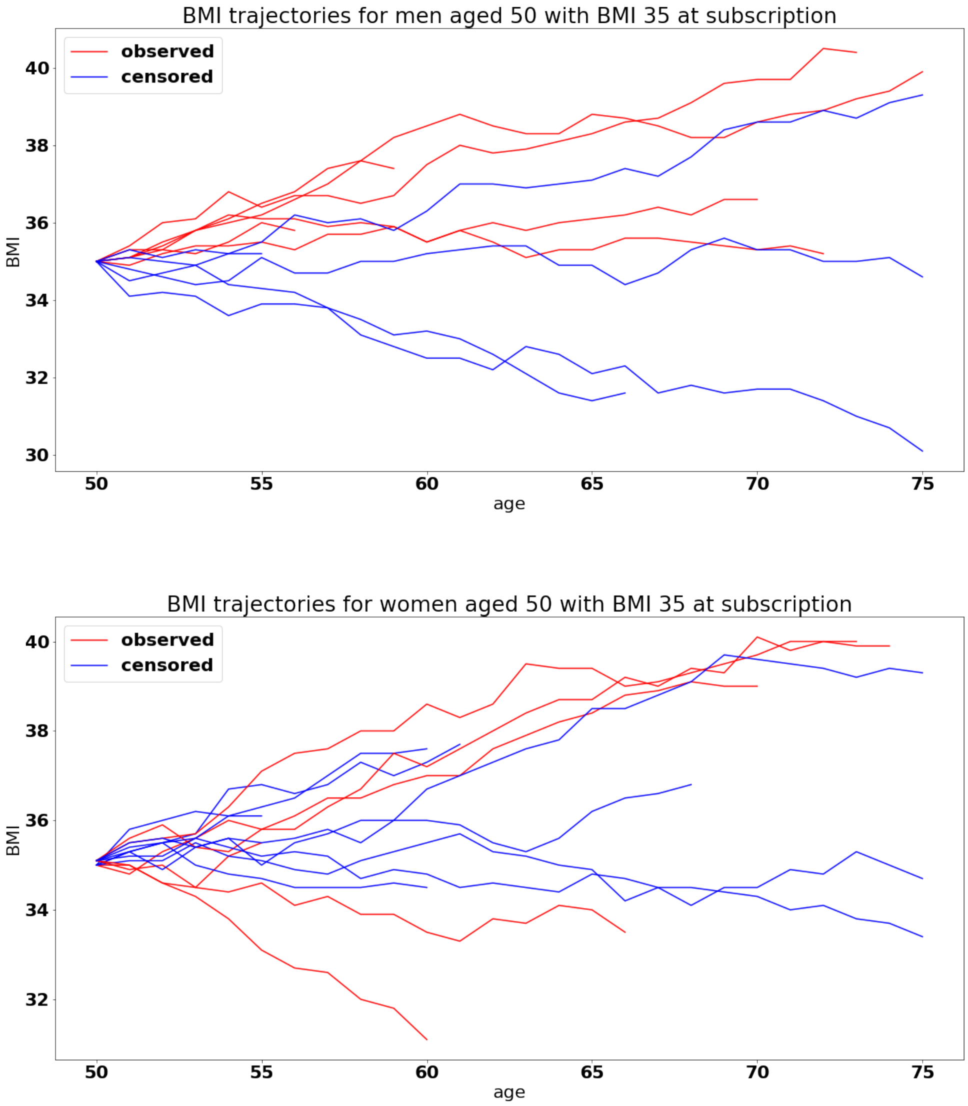


Figure 3.13: BMI trajectories before leaving the database, by gender

Mortality

I provide a brief exploratory survival analysis of my simulated database in order to analyze the impact of risk factors on the survival curves. You can notice on figures 3.14, 3.15, 3.16 that both the theoretical and observed mortality rates are true to the shape of the risk functions implemented for each parameter. It is very apparent on figure 3.14 that the mortality risk is exponential on age ; this is solely due to the

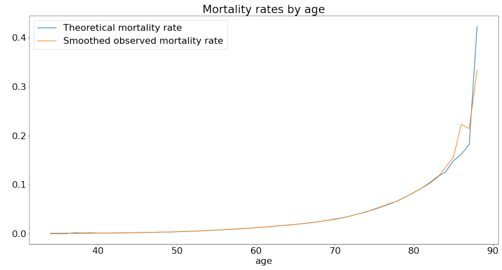


Figure 3.14: Mortality per age

Figure 3.15: Mortality per BMI

underlying INSEE mortality tables, as age is not explicitly taken into account in our risk factor calculation. For BMI (figure 3.15), systolic and diastolic (figure 3.16) blood pressures, the very fluctuating aspect of the mortality towards the high values is due to a low number of observations in these limit zones.

The Kaplan-Meier estimator of the survival curves (figures 3.17 and 3.18) reflects the relationships present in the INSEE mortality tables and my multiplying risk factor, namely that the survival curves are lower for men, for smokers, for obese and for the hypertense.

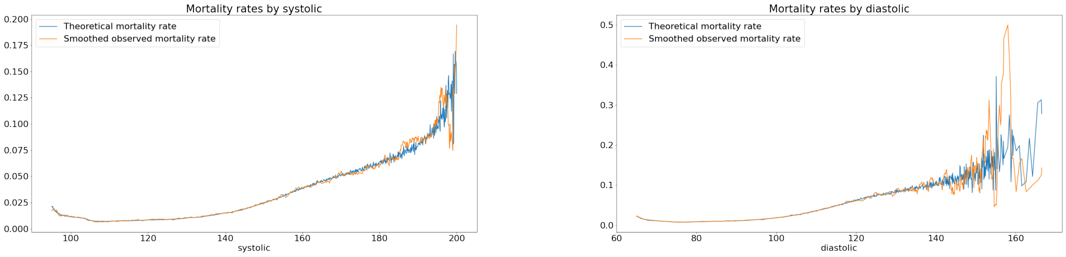


Figure 3.16: Mortality per systolic or diastolic blood pressure

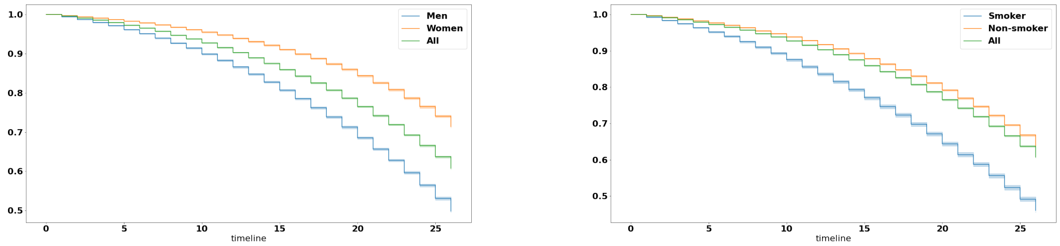


Figure 3.17: Survival curves per gender (left) and smoking habit (right)

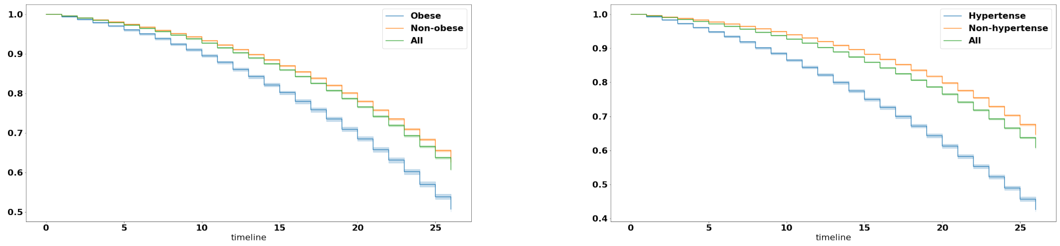


Figure 3.18: Survival curves per obesity (left) and hypertension (right)

3.3.3 Limitations

The limitations to the realism of our simulated cohort are:

1. The mortality is mechanically higher than in a real population. Indeed, our baseline for a "perfect" individual with 0 risk factor is taken as the INSEE 2016 mortality table, where the hazard rates are calculated over the entire French population, thus accounting for a variety of risk profiles.
2. The sources on which our simulation relies are heterogeneous : the mortality and gender ratio data come from INSEE 2016 data, thus representing the French population structure, whereas the parts of smokers, obese and hypertense by age category come from Canada 2019 data.
3. Our initial population is simulated according to overall population statictics, not taking into account the selection bias at subscription of life insurance that would favor initially healthy individuals, or originating from a specific socio-economic level ; that point exacerbates even more point 1.
4. The correlations that have been introduced in the initial distribution of some parameters are completely arbitrary and do not necessarily propagate to the following time steps.

5. The BMI and blood pressure trends are totally arbitrary and not changing, independent of age, gender or initial values. The amplitude of the average evolution and variability on a yearly basis is also arbitrary, constant and independent of age, gender or past values.
6. Our model focuses only on cardiovascular-like risk, whereas mortality over an entire population has much more varied causes. For example, four main causes, all genders combined, account for two-thirds (67.1%) of deaths in France (2015 data): malignant tumors (28.7% of deaths), violent deaths (accidents, suicides, and other external causes of death, 6.6%), circulatory system diseases (cardiovascular diseases, 24.5%), and respiratory system diseases (7.3%).
7. The number of medical factors considered is voluntarily very limited, and lacking crucial sociological information such as the marital status or socioeconomic category for instance. Furthermore, for the represented factors there could be a considerable lack of diversity in the trajectories and correlations observed as they have been simulated based on arbitrary modeling choices.
8. The data, though incomplete, is "too perfect" : there is no missing information, which would be very unrealistic in a real-world medical study.

Chapter 4

Deep Learning in Survival Analysis

4.1 Deep Learning in a word

Deep learning is a class of machine learning algorithms using deep neural networks. They are widely used in natural language processing (NLP), computer vision, reinforcement learning, etc. They are composed of successive combination of scalar products of weights by input features, which are then fed to a non-linearity to increase the representational power (see figure 4.1). The optimization is based on gradient descent on a predefined loss function.

4.1.1 Introduction

Interest of Deep Learning

Deep learning refers to a set of learning methods attempting to model data with complex architectures combining different non-linear transformations. The domain has gained huge interest by the variety of its applications, its adaptivity to all kinds of problems, and its generally state-of-the-art performances on mildly to highly complex problems. Potential applications are very numerous : these techniques have enabled significant progress in the fields of sound and image processing, including facial recognition, speech recognition, computer vision, automated language processing, text classification (for example spam recognition).

Deep learning is based on the use of neural networks, a very flexible framework for modeling exotic distributions, thanks to its high number of parameters (typically in the order of magnitude of the million) and use of more or less complex (depending on the number of layers of the network) combinations of non-linear functions : the elementary bricks of deep learning are these neural networks, that are combined to form the deep neural networks.

There are several possible architectures for neural networks :

- The multilayer perceptrons, also called dense or feed-forward;
- The convolutional neural networks (CNN), used for image problems;
- The recurrent neural networks, for sequential data such as text or time series.

They combine cascade of layers and need clever stochastic optimization algorithms, initialization, and also choice of structure.

History

Deep learning is a class of algorithms that uses multiple layers of densely connected perceptrons[1] to progressively extract higher-level features (vectorial representations in a latent, or hidden, space) from the raw input. For example, in image processing, lower layers may identify texture, while higher layers may identify shapes or edges.

Modern deep learning models are based on artificial neural networks. The word "deep" in "deep learning" refers to the number of layers through which the data is transformed, corresponding to the "complexity" of the model function[2]. Deep models are often able to extract better features than shallow models, hence extra layers help in learning the features effectively and extracting signal efficiently.

Deep learning methods eliminate feature engineering by mapping data to compact hidden representations akin to principal components. What's more, deep learning algorithms can be applied both to supervised or unsupervised learning tasks, which is a huge advantage since unlabeled data are more abundant than labeled data.

4.1.2 Principles

Neurons and Activations

An artificial neuron is a function f_j of the input $x = (x_1, \ldots, x_d)$ weighted by a matrix of connection weights $w_j = (w_{j,1}, \ldots, w_{j,d})$, completed by a neuron bias b_j, and associated to an activation function ϕ, namely

$$y_j = f_j(x) = \phi(< w_j, x > + b_j)$$

Figure 4.1 illustrates the general structure of an artificial neuron.

---

[1] The perceptron is a function that maps its input x to an output value $f(wx+b)$, where w and b are the weights and biases matrices optimized through back-propagation, and f a differentiable non-linear activation function, see figure 4.1.

[2] This notion of complexity of the model function is quantified through the VC-dimension, but we will not address this here.

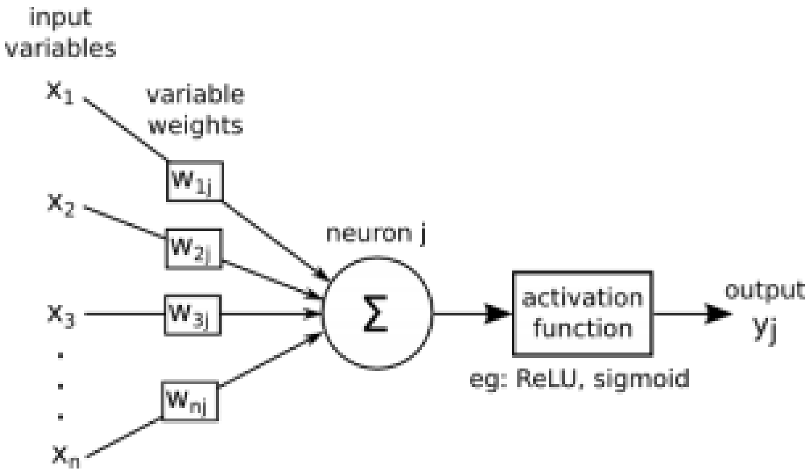


Figure 4.1: General representation of an artificial neuron (Perceptron)

Several activation functions can be considered :

- The identity function $\phi(x) = x$: the utility of this function is very limited since $f_j(x)$ is then a simple linear transformation of x, rendering the use of more than 1 layer unnecessary : an arbitrary number of successive linear transforms is equivalent to one. Therefore, the activation function ϕ is usually taken non-linear.

$$A_1 \times (A_2 x + b_2) + b_1 = (A_1 \times A_2)x + (A_1 b_2 + b_1)$$

- The sigmoid, or logistic, function $\phi(x) = \frac{1}{1+exp(-x)}$: it is a strictly increasing, differentiable function that maps inputs to the $]0,1[$ range, with $\phi(0) = 1/2$;
- The hyperbolic tangent, or tanh, function $\phi(x) = \frac{exp(x)-exp(-x)}{exp(x)+exp(-x)}$: it is a strictly increasing, differentiable function that maps inputs to the $]-1,1[$ range, with $\phi(0) = 0$;
- The hard-threshold function $\phi_\beta(x) = \mathbb{1}[x \geq \beta]$: very often, β is taken equal to 0, but it can also be an optimized hyperparameter. This activation function is less used due to its lower representativity power ;
- The Rectified Linear Unit, or ReLU, function $\phi(x) = max(0,x)$: this is a hugely popular method since, contrarily to the sigmoid and tanh, it doesn't have an upward bound on its values. This function is not differentiable in 0 but in practice this is not really a problem since the probability to have an entry exactly equal to 0 is generally null. It does induce a so-called dead ReLU problem, related to the vanishing gradient issue, which we will address later ; to curb this limitation, it has a number of variants :
    - Gaussian Error Linear Unit (GELU) : $\phi(x) = x.\Phi(x)$, where Φ is the cumulative

distribution function of the gaussian distribution ; this is the default activation in BERT, a state-of-the-art NLP model ;

- Sigmoid Linear Unit (SiLU) : $\phi(x) = x.\sigma(x)$, σ the sigmoid function ;
- SoftPlus or SmoothReLu : $\phi(x) = ln(1 + e^x)$;
- Exponential Linear Unit (ELU) : $\phi(x) = \begin{cases} x & \text{if } x > 0, \\ a(e^x - 1) & \text{otherwise} \end{cases}$

A usual value for a is 1, with 0 this is just the standard ReLU ;

- Parametric ReLU : $\phi(x) = \begin{cases} x & \text{if } x > 0, \\ ax & \text{otherwise} \end{cases}$

When a is a positive, arbitrarily small ϵ, this activation is designated as Leaky ReLU.

Figure 4.2 shows the general behaviour of these most common activation functions, and figure 4.3 compares the different ReLU variants.

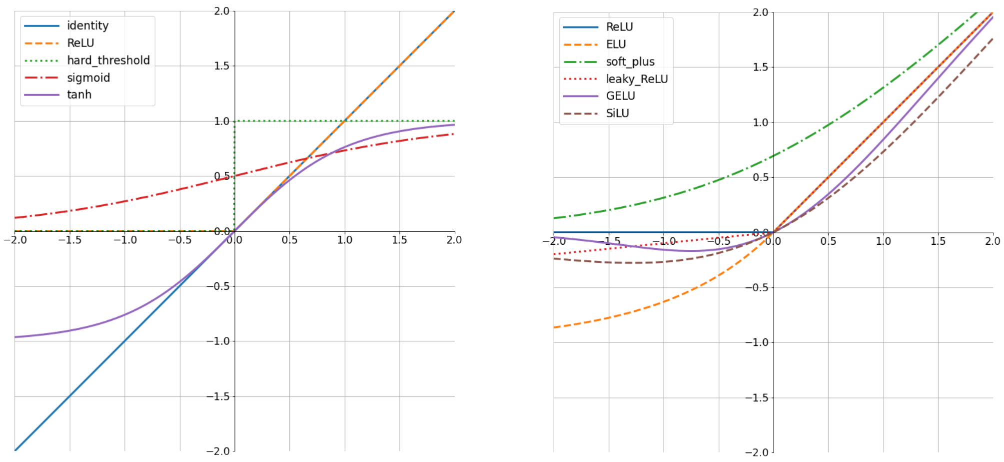


Figure 4.2: Common activations

Figure 4.3: ReLU variants

Figure 4.4: Most common activation functions

4.1.3 Gradient Descent and Back-Propagation

Neural networks function by optimizing their set of parameters, i.e. the weights and biases of all layers, $\theta = (W^{(1)}, b^{(1)}, \ldots, W^{(L+1)}, b^{(L+1)})$ with L the number of layers in the network. The main objective of the model function f is, for each input data X_i, to predict the corresponding target Y_i as accurately as possible.

To that purpose, we have to define a loss function $\mathscr{L}$ that is a metric of the error made by the model (i.e. the distance between the prediction $f(X_i)$ and the corresponding target Y_i). There are a number of different loss functions, depending on the nature of the problem (regression or classification) and the objective of the learner (ranking problem, average accuracy, minimize extreme errors, etc). This loss function is what we seek to optimize : we want to minimize (or maximize in case of an adequation metric) the error of the model on the data is is confronted with.

The main requirement about the chosen loss function is that it must be differentiable : in other words, we should be able to compute a gradient of the loss with respect to the parameters of the model. Indeed, this gradient is the local slope of the loss function, it gives the general direction to follow to maximize the objective function (the opposite direction if you want to minimize it).

Some common loss functions are :

- Mean Squared Error : $\mathscr{L}_{MSE}(\hat{Y}_B, Y_B) = \frac{1}{|B|} \sum_{i \in B} (\hat{Y}_i - Y_i)^2$
- Mean Absolute Error : $\mathscr{L}_{MAE}(\hat{Y}_B, Y_B) = \frac{1}{|B|} \sum_{i \in B} |\hat{Y}_i - Y_i|$

In the case of classification :

- For binary classification : Cross-Entropy Loss :

$$\mathscr{L}_{CE}(\hat{Y}_B, Y_B) = \frac{1}{|B|} \sum_{i \in B} Y_i ln(\hat{Y}_i) + (1 - Y_i) ln(1 - \hat{Y}_i)$$

- For multi-class classification : Multi-class Cross-Entropy Loss :

$$\mathscr{L}_{MCCE}(\hat{Y}_B, Y_B) = \frac{1}{|B|} \sum_{i \in B} \sum_{k=1}^{K} Y_i^{(k)} ln(\hat{Y}_i^{(k)})$$

  where k is the number of classes ;

In the case of a ranking problem :

- Concordance index
- etc.

The stochastic gradient descent algorithm functions as follows :

Initialization : All parameters $\theta = (W^{(1)}, b^{(1)}, \dots, W^{(L+1)}, b^{(L+1)})$, with L the number of layers in the network, are initialized ;

For N iterations :

For each batch B of training data $(X_i, Y_i)_{i\in B}$:

$$\theta = \theta - lr\frac{1}{m}\sum_{i\in B}\nabla_\theta \mathscr{L}(F(X_i), Y_i)$$

Here, lr is what we call a learning rate : it is the step size that is taken in the direction of the negative gradient.

$$\theta = \theta - lr\frac{1}{m}\sum_{i\in B}\nabla_\theta \mathscr{L}(F(X_i), Y_i)$$

Very often, a penalization term is added, also called l2 regularization or weight decay, to ensure that the model doesn't attribute too big weights to some parameters, thus working on preventing overfitting. The lr term, called learning rate, corresponds to the size of the step you take in the direction descending the gradient.

The vast majority of successful deep neural networks are trained using variants of stochastic gradient descent (SGD) algorithms. Recent attempts to improve SGD can be categorized into two approaches :

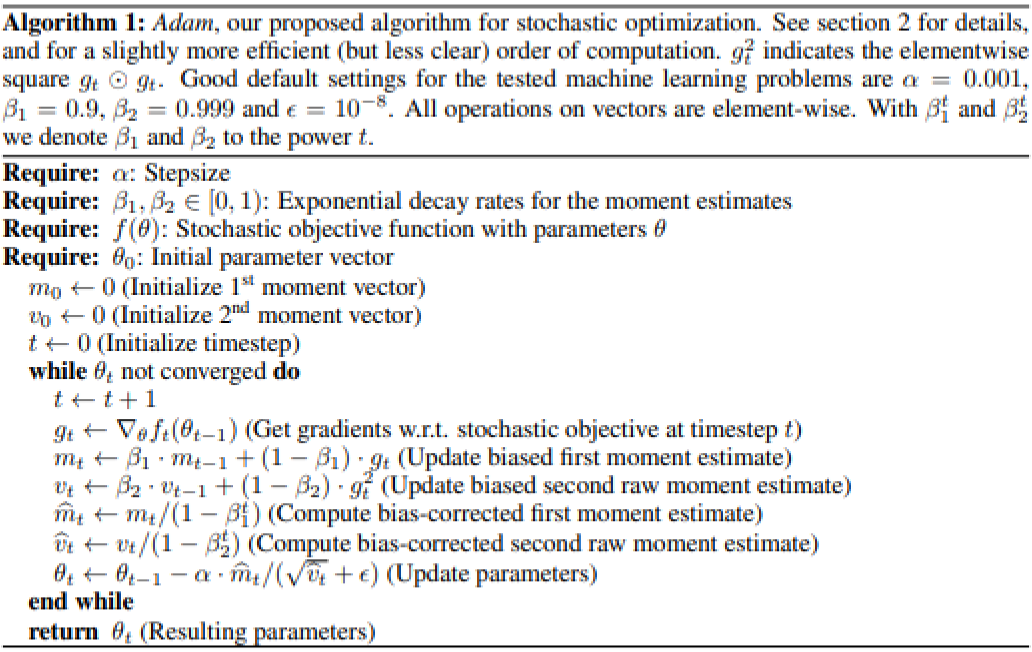
1. **Adaptive learning rate** schemes, such as AdaGrad and Adam,
2. **Accelerated** schemes, such as heavy-ball and Nesterov momentum.

4.1.4 The Adam Optimizer

The Adam optimization algorithm is an extension to stochastic gradient descent that has seen broader adoption for deep learning applications in computer vision and natural language processing. Empirical results demonstrate that Adam works well in practice and compares favorably to other stochastic optimization methods.

Adam (*ADAptative Moment estimation*) is based on adaptive estimates of lower-order moments (first and second momentum). The method is very easy to implement, computationally efficient, and is well suited for problems that are large in terms of data and/or parameters. Adam can also be used in problems with very noisy or sparse gradients. The hyper-parameters have intuitive interpretations and most of the time require little tuning.

Adam is different from classical stochastic gradient descent algorithms : indeed, stochastic gradient descent maintains a single learning rate α for all weight updates and the learning rate does not change during training. A learning rate is maintained for each network weight (θ) and separately adapted during the learning process. However, Adam combines the advantages of two other extensions of SGD:



Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

```
Require: $\alpha$: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters $\theta$
Require: $\theta_0$: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1st moment vector)
  $v_0 \leftarrow 0$ (Initialize 2nd moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while $\theta_t$ not converged do
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep $t$)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return $\theta_t$ (Resulting parameters)
```

Figure 4.5: The Adam algorithm for stochastic optimization

1. **Adaptive Gradient Algorithm (AdaGrad)** that maintains a per-parameter learning rate, therefore improving performance on problems with sparse gradients ;
2. **Root Mean Square Propagation (RMSProp)** that also maintains per-parameter learning rates, that are updapted based on the average of recent magnitudes of the gradients for the weight. This is beneficial on noisy, non-stationary problems.

The Adam algorithm construction is illustrated on figure 4.5. Let $f(\theta)$ be the objective function (loss function, depending on the weights θ of the model).

$f(\theta)$ is assumed to be a stochastic scalar function that is differentiable with respect to the parameters θ (weights). We are interested in minimizing the expected value of this function, $\mathbb{E}[f(\theta)]$ with respect to its parameters θ. We call $f_1(\theta), ..., f_T(\theta)$ the realisations of the stochastic function at timesteps $1, ..., T$ of the training epoch. The stochasticity stems from the evaluation at random subsamples (minibatches) of datapoints : for each time step t, the function is not evaluated on the same batch of data. With $g_t = \nabla_\theta f_t(\theta)$ we denote the gradient, i.e. the vector of partial derivatives of f_t, w.r.t θ evaluated at timestep t.

The algorithm updates exponential moving averages of the gradient (m_t) and the squared gradient (v_t) where the hyper-parameters $\beta_1, \beta_2 \in [0, 1)$ control the exponential decay rates of these moving averages. The moving averages themselves are estimates of the first moment (the mean) and the second raw moment (the uncentered variance) of the gradient. We have to be careful here that the second moment is not the variance

$\mathbb{E}\left[(g_t - \mathbb{E}(g_t))^2\right]$ of the gradient, but the squared gradient $\mathbb{E}\left[g_t^2\right]$, which is more tricky to interpret.

These moving averages are initialized as vectors of 0's, leading to moment estimates that are biased towards zero, especially during the initial timesteps, and especially when the decay rates are small (i.e. the β are close to 1). The good news is that this initialization bias can be easily counteracted, resulting in bias-corrected estimates $\hat{m}_t$ and $\hat{v}_t$.

It is interesting to analyse the relationship between the hyperparameter β_1 (first moment) and dropout effect. This parameter modules how much importance you put on past gradients, compared with the newly calculated gradients, when computing the update for a weight. The higher β_1, the more importance you put on past gradients, and the less sensible you are to brutal fluctuations of the gradients. It is noticeable that the effects of dropout and variations of β_1 compensate each other : the higher p, the more fluctuating $\hat{m}_t$ is, the higher β_1, the less fluctuating $\hat{m}_t$ is ; p and β_1 therefore have compensating effects.

4.2 Deep Learning for Sequential Data

4.2.1 Recurrent Neural Networks

In what precedes, we have seen that deep learning has recently been introduced in the tools library available for survival analysis. In this spirit, it makes sense to consider using dynamic deep learning models to account for the effect of time-varying covariates : they are very common and have been hugely developed mainly for natural language processing (NLP) problems, where sequentiality (the order of words in a sentence) is fundamental to the understanding of the problem.

Recurrent neural networks (RNNs) are the state of the art algorithm for sequential data and are used in all kinds of applications from time-series financial data to natural language processing. It is the first algorithm that remembers its input, due to an internal memory, which makes it perfectly suited for machine learning problems that involve sequential data. It is one of the algorithms behind the scenes of the amazing achievements seen in deep learning over the past few years. In this part, we'll cover the basic concepts of how recurrent neural networks work, what the biggest issues are and how to solve them.

RNNs are a powerful and robust type of neural network, and belong to the most promising machine learning algorithms in use because it is the only one with an internal

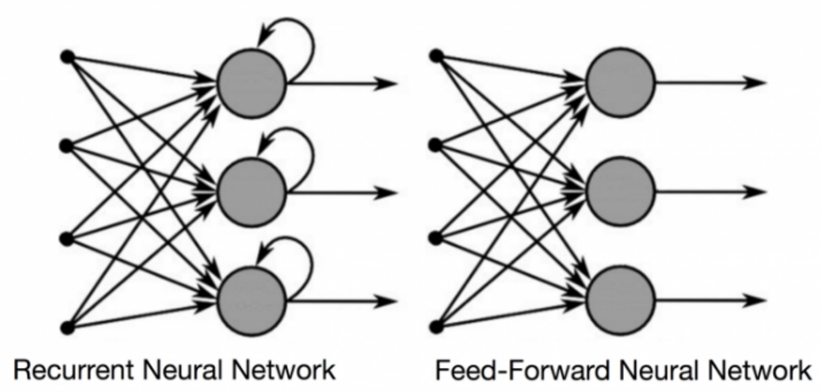


Figure 4.6: Recurrent and feed-forward architectures

memory. Like many other deep learning algorithms, recurrent neural networks were initially created in the 1980's, but only in recent years the humungous increase in computational power along with the the massive amounts of data have they become widely used. Long short-term memory (LSTM), that solve the vanishing gradient issue, were introduced in the 1990's.

Because of their internal memory, RNNs can remember important things about the input they received, which allows them to be very precise in predicting what's coming next. This is why they're the preferred algorithm for sequential data like time series, speech, text, financial data, audio, video, weather and much more. Recurrent neural networks can form a much deeper understanding of a sequence and its context compared to other algorithms.

RNNs and feed-forward neural networks get their names from the way they channel information. In a feed-forward neural network, the information only moves in one direction : from the input layer, through the hidden layers, to the output layer. The information moves straight through the network and never touches a node twice (see figure 4.6). Feed-forward neural networks have no memory of the input they receive and are bad at predicting what's coming next. Because a feed-forward network only considers the current input, it has no notion of order in time. It simply can't remember anything about what happened in the past except its training.

In a RNN on the other hand, the information cycles through a loop (see figure 4.6). When it makes a decision, it considers the current input and also what it has learned from the inputs it received previously. A usual RNN has a short-term memory : this is due to the vanishing gradient problem inherent to the structure of the RNN. This is easy to see if you consider the usual activation functions of neural networks :

- **sigmoid** : $\frac{d\sigma(x)}{dx} = \sigma(x)(1-\sigma(x))$
- **hyperbolic tangent** : $\frac{dtanh(x)}{dx} = 1 - tanh^2(x)$

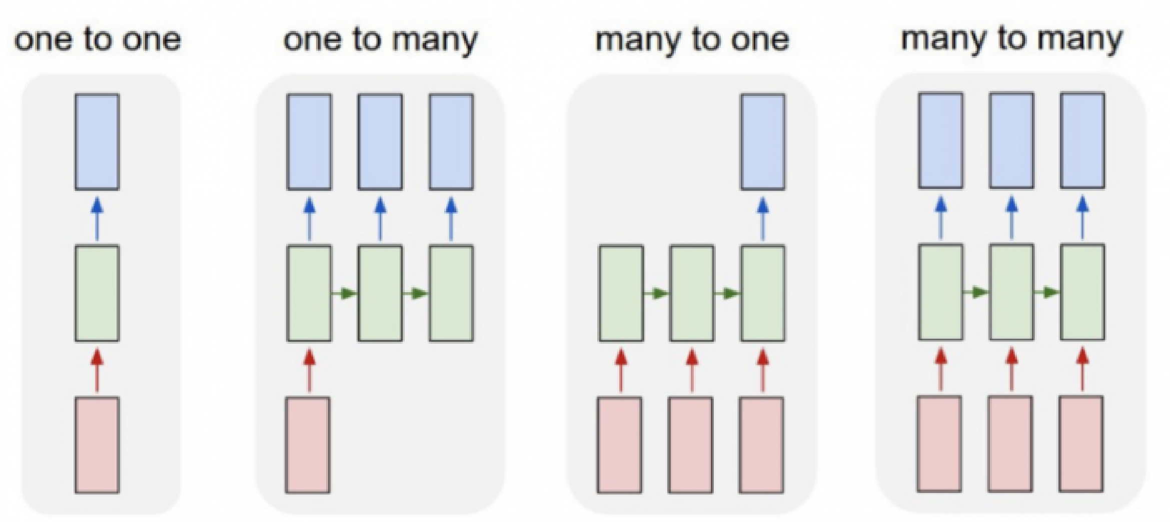


Figure 4.7: Mapping structures for RNNs

- **ReLU** : $\frac{dR(x)}{dx} = \mathbb{1}\left[x > 0\right]$

Therefore, you can notice that all have a norm always comprised between 0 and 1, therefore the chain rule in the back-propagation of gradients will make these norms multiply at each time step and the gradient will mechanically quickly vanish after a few time steps. To solve this issue, the LSTM have a long-term memory (more on that later).

Therefore, a RNN has two inputs: the present and the recent past. This is important because the sequence of data contains crucial information about what is coming next, which is why a RNN can do things other algorithms can't. A feed-forward neural network assigns, like all other deep learning algorithms, a weight matrix to its inputs and then produces the output. Note that RNNs apply weights to the current and also to the previous input. Furthermore, a recurrent neural network will also tweak the weights for both through gradient descent and back-propagation.

Also note that while feed-forward neural networks map one input to one output, RNNs can map one to many, many to many (translation) and many to one (classifying a voice), see figure 4.7. This will be particularly useful in our case, since we want to predict either a death probability at each time step, or an overall risk score, for a sequence of observed medical covariates (bmi at different time steps for example) or a single initial observation.

4.2.2 Long Short-Term Memory

Long short-term memory networks (LSTMs) are an extension for recurrent neural networks, which basically extends the memory. Therefore it is well suited to learn from important experiences that have very long time lags in between (as in our case, where

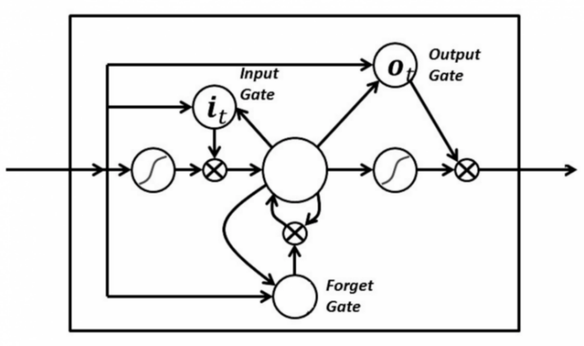


Figure 4.8: An illustration of an LSTM cell

survival durations are often extended over years or decades).

The units of an LSTM are used as building units for the layers of a RNN, often called an LSTM network. LSTMs enable RNNs to remember inputs over a long period of time. This is because LSTMs contain information in a memory, much like the memory of a computer. The LSTM can read, write and delete information from its memory.

This memory can be seen as a gated cell, with gated meaning the cell decides whether or not to store or delete information (i.e., if it opens the gates or not), based on the importance it assigns to the information. The assigning of importance happens through weights, which are also learned by the algorithm. This simply means that it learns over time what information is important and what is not.

In an LSTM you have three gates: input, forget and output gate. These gates determine whether or not to let new input in (input gate), delete the information because it isn't important (forget gate), or let it impact the output at the current timestep (output gate). On figure 4.8 is an illustration of an LSTM cell with its three gates.

4.3 Survival deep learning

Over the past years, a significant amount of research in machine learning has been conducted in combining survival analysis with neural networks. With the development of deep learning technologies and computational capacities it is possible to achieve outstanding results and implement a range of architectures on very large datasets with different underlying methodologies and more specific learning inside.

Deep neural networks used in survival analysis can be split into two main categories : Cox-based networks, that optimize variants of the Cox-likelihood as their loss functions, and other models, that include a very classic binary classification model (survives / dies) by optimizing the well-known binary cross-entropy loss, but also more advanced models based on individual risk assessment with ranking metrics, recurrent models, etc.

4.3.1 Cox Proportional Hazards Models

The initial adaptation of neural networks to a survival analysis problem (Farragi and Simon, 1995) was based on the generalization of the Cox proportional hazards model with only a single hidden layer. The main focus of the initial model was to learn relationships between primary covariates and the corresponding hazard risk function. Following development of the neural network architecture with Cox regression proved that in real-world large datasets with non-linear interactions between variables it is rather complicated to keep the main proportionality assumption of Cox regression model. However, Farragi and Simon's network extended this non-linearity quality.

The traditional Cox regression consists in optimizing the Cox partial likelihood. This likelihood is defined with the following formula with parametrized weights β:

$$L(\beta) = \prod_{i:e_i=1} \frac{exp(\hat{h}_\beta(x_i))}{\sum_{j\in R(t_i)} exp(\hat{h}_\beta(x_j))}$$

where t_i, e_i, x_i are time, event, baseline covariate data in the i-th observation respectively. More explicitly, this is a product of probabilities at the time t_i for the i-th observation given the set of risk individuals (R) that are not censored and have not experienced the event of interest before time t_i.

Cox neural networks are Cox models where the neural network implements the function g_β, and the losses used are variants (with penalization, log, etc) of the Cox partial likelihood.

DeepSurv In the 2016 paper *DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network*, Katzman et al. proposed a more sophisticated deep learning architecture named DeepSurv. DeepSurv is an extension of Simon-Farragi's model with more hidden layers and use modern techniques for controlling the gradient descent. Similar to Katzman et al.'s feed-forward neural network, the model estimates each individual's characteristics on their hazard rates.

$$l(\theta) = -\sum_{i:E_i=1} (\hat{h_\theta}(x_i) - log \sum_{j\in R(T_i)} e^{\hat{h_\theta}(x_i)})$$

Generally, the structure of this neural network is quite straightforward. Comparing to Simon-Farragi network, DeepSurv is a configurable with multiple number of hidden

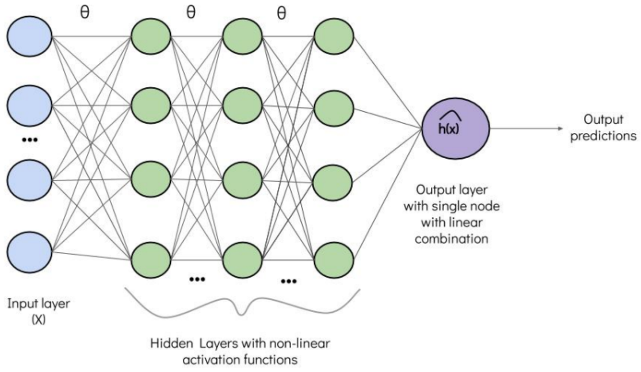


Figure 4.9: Tensor view

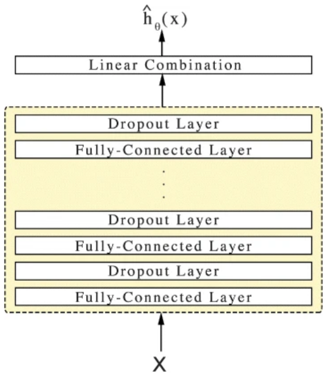


Figure 4.10: Layer view

Figure 4.11: DeepSurv architecture

layers. It showed improvements of the Cox Proportional Hazards model and the performance metrics when dealing with non-linear data. In addition to that, while estimating the log-risk function $h(X)$ with the Cox Proportional Hazards model they used the linear combination of static features from given data X and the baseline hazards. The main impact of DeepSurv was to drop the proportional hazards assumption out.

The input data X is represented as a set of observed covariates. Hidden layers in this model are fully-connected nonlinear activation layers with not necessarily the same number of nodes in each of them, followed by dropout layers to avoid overfitting (see previous introduction to deep learning). The output layer has only one node with a linear activation function which gives the output $\hat{h}_\theta$ (log-risk hazard estimations) (see figure 4.11).

4.3.2 Static Architectures

Any deep learning architecture can be adapted to modeling survival durations (regression problem) or survival at a given term (binary classification problem). A notable one is *DeepHit*, presented hereunder, but there are a wide variety of other models.

DeepHit *DeepHit* is a deep neural network for survival analysis, introduced by Changhee Lee, William R. Zame, Jinsung Yoon and Mihaela van der Schaar in 2018. Its specificity relies in dealing with competing risks[3].

The goal is to estimate the joint distribution of the first hitting time and competing events. DeepHit is a multi-task network (Collobert and Weston 2008) which consists of

---

[3] A competing risk is an event whose occurrence masks the outcome of the primary event of interest: it is equivalent to censorship. For example, in a study examining time to death attributable to cardiovascular causes, death attributable to non-cardiovascular causes (e.g. a car accident, prostate cancer, etc) is a competing risk. Competing risks occur frequently in the analysis of survival data, particularly when durations are quite extended in time.

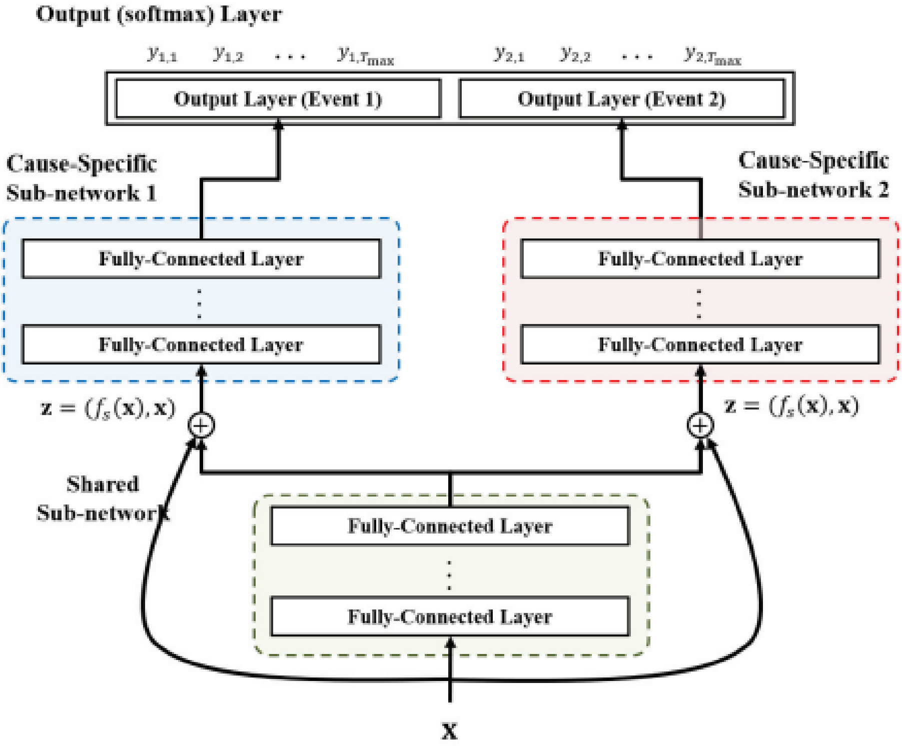


Figure 4.12: DeepHit Architecture

a shared sub-network and N cause-specific sub-networks. The architecture differs from a conventional multi-task network:

- a single softmax layer as the output layer, to ensure that the network learns the joint distribution of N competing events and not the marginal distributions of each event ;
- a residual connection from the input covariates into the input of each cause-specific sub-network, jumping over the 2 feed-forward layers, so that the actual covariates and their embeddings are concatenated and provided as an input to each sub-network.

Competing risks could prove very useful in an insurance context, since causes of death are numerous, some can be highly correlated (e.g. heart attack and stroke are both cardiovascular-related causes of death) and others independent (e.g. breast cancer and Alzheimer disease) or even non-medically related (e.g. car accident, murder). They could also be of great help in prevention about behaviors that create multiple competing risks, e.g. eating junk food.

4.3.3 Recurrent Deep Learning Models

As an alternative approach, fully parametric survival models use Recurrent Neural Network to sequentially predict a distribution over the time to the next event: RNN-Surv (2018), Weibull Time-To-Event RNN (2017), DRSA (2018) etc.

Contrary to Cox-based models, which assume an invariant structure of the risk through time, recurrent neural networks are able to model time varying effects. To be applied, the time of each individuals must be discretized into small intervals.

The use of recurrent models is particularly interesting in life insurance, where we want to be able to use updates on a patient's health (for example, whether he stopped smoking, or gained 20kg) to update the predictions on his survival probabilities.

Time varying covariates can be very useful to handle new business usecases in insurance and provide preventive health services. With the increase of IoT devices on the market (connected weight scale, smart watch, ...) insurers will be able to access more covariates that evolve through time. A deep understanding of RNN models will help insurers to better model the dynamic of the portfolio through time.

Weibull Time To Event RNN

The *Weibull-Time-To-Event-RNN* is a framework for prediction of the time to the next event applicable when we have time-varying covariates or time series of varying lengths. The model estimates the distribution of time to the next event as having a discrete or continuous Weibull distribution with parameters being the output of a recurrent neural network. The model is trained using a special objective function (log-likelihood-loss for censored data) commonly used in survival analysis.

The Weibull Distribution : The Weibull distribution is a continuous probability distribution, first identified by Fréchet (1927). It is named after Swedish mathematician Waloddi Weibull who described it in 1951.

It is defined by its cumulative distribution function :

$$F(x) = \mathbb{1}\left[x \geq 0\right]\left[1 - exp^{-\left(\frac{x}{\lambda}\right)^k}\right]$$

where $\lambda > 0$ and $k > 0$ are respectively the scale and shape parameters. The probability density functions derived from the cumulative distribution, for different values of the scale and shape parameters, are pictured in figure 4.13.

More particularly, one can note that :

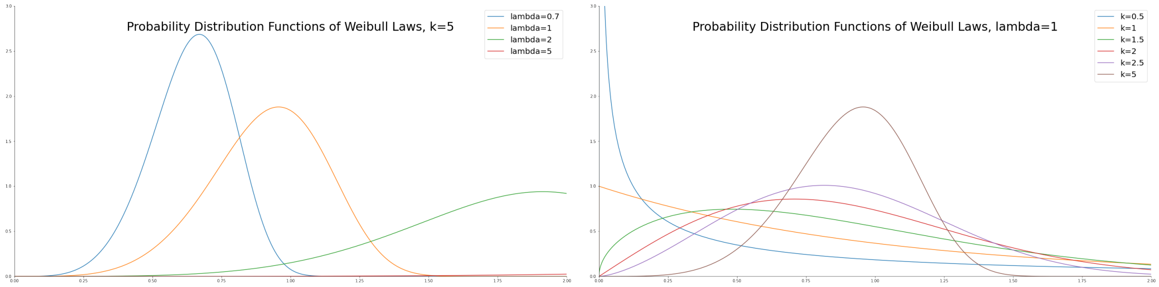


Figure 4.13: Probability Distribution Functions of Weibull Laws

- $k < 1$: the mortality rate decreases over time, for example if there is significant infant mortality and the "fragile" individuals are progressively weeded out of the population.
- $k = 1$: the mortality rate is constant over time, this might suggest that death is caused by random external events : indeed, we fall back on an exponential distribution, a no-memory law.
- $k > 1$: the mortality rate increases with time : there is an "aging" process.

Deep Recurrent Survival Analysis

Deep Recurrent Survival Analysis, introduced in 2018 by Kan Ren, Jiarui Qin, Lei Zheng, Zhengyu Yang, Weinan Zhang, Lin Qiu and Yong Yu, is very similar to *RNN-Surv*, the main object of this study, with 2 main differences :

- The use of a RNN instead of a LSTM ;
- The prediction of the hazard rates instead of the survival probabilities.

I will not detail its structure as it is identical to RNN-Surv.

4.4 RNN-SURV: a Deep Recurrent Model for Survival Analysis

The main focus of this actuarial thesis has been to implement and improve a model presented in the paper *RNN-SURV: a Deep Recurrent Model for Survival Analysis* by Eleonora Giunchiglia, Anton Nemchenko, and Mihaela van der Schaar (2018).

4.4.1 Model

The authors place themselves in a discrete time frame, dividing the maximal observed time into K intervals $(t_0, t_1]$, . . . ,$(t_{K1}, t_K]$. They also assume that the characteristic function modeling T_i is constant within each interval $(t_{k1}, t_k]$ with $k = 1, ..., K$, meaning

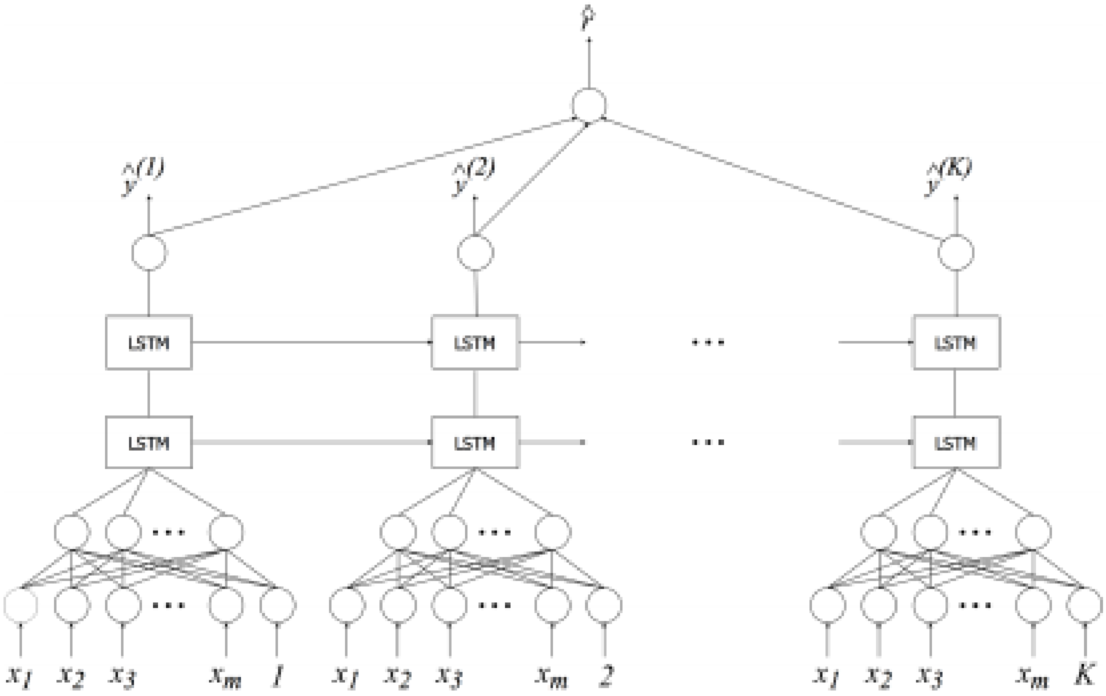


Figure 4.14: RNN-SURV with 2 feedforward layers, followed by 2 recurrent layers.

that the study focus on the mortality per interval, and supposes the risk uniform within the time interval. Given a patient i, the purpose of the model is to output both an estimate $\hat{y}_i^k$ of the survival probability y_i^k for the k^{th} time interval:

$$y_i^k = \mathbb{P}(T_i > k | T_i \geq k)$$

and an estimated risk score $\hat{r}_i$ approximating the individual's relative risk r_i:

$$\forall i, j, \quad T_i < T_j \iff r_i > r_j$$

The model's architecture is depicted in figure 4.14. There are K inputs, which are vectors of the covariates at each time step $t = 1...K$, concatenated with the the time value. These inputs are fed to what the authors call **embeddings layers**, which are feedforward layers aiming at representing the vector of covariates in a latent space. This latent representation becomes the input, at each time step, of the LSTM.

Several remarks are necessary on this architecture :

- The feedforward layers are called *embeddings* because they output a vectorial representation in a latent-space, lower dimensional than the inputs, but they are not strictly speaking embeddings in an NLP sense since they are not pretrained on an unsupervised problem to build pretrained vectorial representations. In NLP, the input space is by definition discrete (words), whereas here our input can be continuous on some covariates (continuous values like blood pressure for instance), therefore the latent space representations are also partially continuous.

- Here, the so-called embeddings layers are optimized during training with the rest of the network, that is not the case in a NLP framework where the embeddings layers are first pretrained, and then frozen during training of the objective model.

The covariates must be provided at each of the K timesteps. However, they may not be observed at each period :

1. In the general case, it is very hard to have regular observations of medical covariates for all patient across time. Most of the time, relevant medical features are only measured when a patient has a medical appointment, which is often very irregular, and not all of them (BMI, blood pressure...) are systematically measured at each appointment.
2. The model takes a input a fixed number of K covariates vectors at K time steps, but some patients may die before reaching the K^{th} period, or leave the study for some reason (stopped showing us to doctor's appointments, changed region, died of some other cause than the one being studied...)

In the first case, there are several ways to fill values for periods where we have no direct observation of the covariates :

- A forward fill of the last observed value ;
- A backward fill of the next observed value ;
- An interpolation of the last and next observed value.

I have chosen to implement the first strategy, for 2 reasons :

- The second and third options imply some kind of *data leakage*, that is to say, a partial view of the future that is somehow making the model "cheat" when asked to predict the future ;
- It is the only one implementable on real-time data.

In the second case, we encounter a very common problematic in NLP where sequences (sentences) do not all have the same length. If the sequence is shorter, it is filled to the maximal length with a *padding vector*, an embedding of a token that is not a word of the dictionnary and that has been pretrained to represent the $< PAD >$ token, a blank filling character. In the case of survival analysis, we have 2 options :

- As in NLP, introduce a $< PAD >$ vector of covariates that is representative of no actual values of the features ; the main issue here will be that contrarily to NLP, we cannot pretrain this embedding vector so that the embeddings layers can learn their own representation, we have to choose arbitrary values
- As before, just forward fill the last observed vector of covariates.

An implementation of both methods will show no striking difference in the results. The only notable fact is that, in the case of the $< PAD >$ vector, categorical data is one-hot encoded using as many indicator features as classes, which is redundant since they sum to 1 the bias of each neuron already accounts for a constant.

As we need to differentiate a padding input from an actual input, we will add a feature that is an indicator is_pad, meaning that at each time step is equals 0 if the individual is still present in the database and 1 if it is not (censored or dead).

Finally, to account for the time-evolving aspect, the authors concatenate the vector of covariates with k the period number or date. Here, I took the liberty to not add this period number, and simply rely on the moving age of the individual (increasing by 1 at each time step) and the bias parameter of the LSTM which, thanks to the recursivity, should bear information of the seniority of the individual in the observations (is it the first time step we oberve him at age 43, or is it actually the 10th timestep and he entered at age 34 ?).

4.4.2 The Concordance Index

One of the most popular performance measures for assessing learned models in survival analysis is the Concordance Index. The C-index is a good indicator of the ranking performance of a model.

Ranking Problem

The general idea when using the concordance index is that we're not interested in the actual predictions, but rather at how our model orders the individuals. For example, we don't want to know if someone will die at time step 5 or 6, but rather if he will die before or after some other individuals.

Casting survival analysis as ranking problem is an elegant way of dealing not only with the typically skewed distributions of survival times, but also with the censoring of the data: Two subjects' survival times can be ordered not only if *both of them are uncensored* but also if *the uncensored time of one is smaller than the censored survival time of the other*. This can be visualized by means of an order graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where the set of vertices $\mathscr{V}$ represents all the individuals, censored or not.

$$\text{Existence of an edge } \mathscr{E}_{ij} \text{ is equivalent to having } \delta_i = 1, Y_i \leq Y_j$$

An edge cannot originate from a censored point, but it can end there. In other words, each oriented edge represents the relationship *dies before.*

If we denote $\mathscr{C}$ the set of edges $\mathscr{E}_{ij}$ of a given graph, we have

$$\mathscr{C} = \{(i,j) | \delta_i = 1 \land (Y_i \leq Y_j)\}$$

In the previous example, we have :

$$\mathscr{C} = \{(A, A|B|C|D|E|G|H); (B|H, B|C|D|E|G|H); (D, G|E); (E, E)\}$$

It is important to note that the pairs of $\mathscr{C}$ are *ordered.* Intuitively speaking, they can be understood as a mapping, for all *uncensored* individuals (for whom we have complete observations), to *all the individuals who live at least as long as them* (no matter if the remaining survival time is observed in its entirety or censored at some ulterior point). This is coherent with the purpose of a ranking metric : we want to compare an individual for whom we have a death time to all the individuals known to have outlived him, and check if the ordering predicted by the model is compatible.

Example

The following cohort can be represented by the order graph in figure 4.15.

i	A	B	C	D	E	F	G	H
Y_i	1	2	3	4	5	0	4	2
δ_i	1	1	0	1	1	0	0	1

The Concordance Index

The evaluation will therefore be based on pairs of individuals : for all individuals i, j, we will check whether our predictions *order* them similarly to the ground truth, or not. That gives us the following definition of the metric :

$$C = \frac{1}{|\mathscr{E}|} \sum_{\mathscr{E}_{i,j}} \mathbb{1}\left[f(x_i) \leq f(x_j)\right]$$

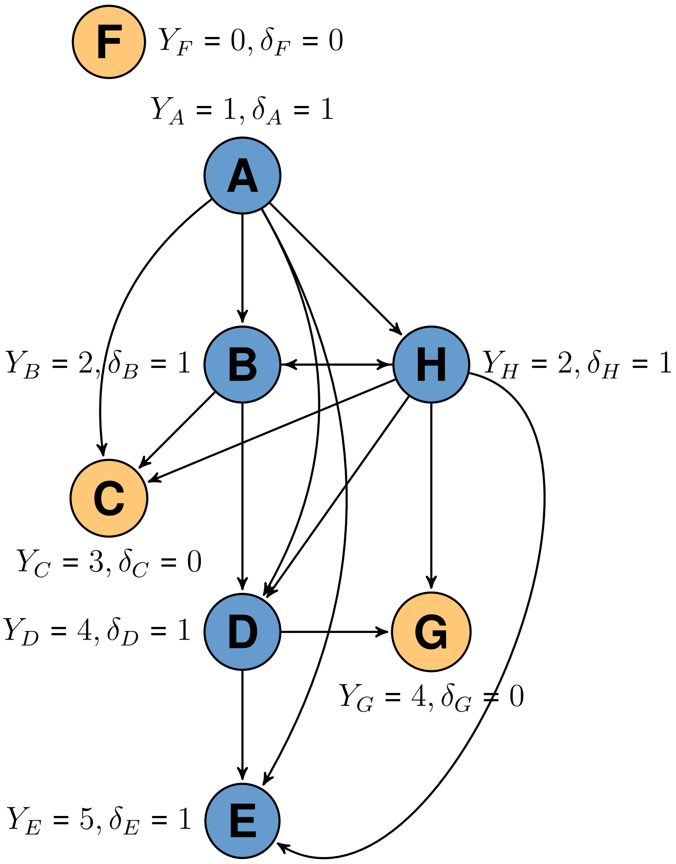


Figure 4.15: Order graph for the example cohort

Or, to write it explicitly without referencing to the order graph :

$$C = \frac{\sum_{i,j} \delta_i \mathbb{1}\left[Y_i \leq Y_j\right] \mathbb{1}\left[f(x_i) \leq f(x_j)\right]}{\sum_{i,j} \delta_i \mathbb{1}\left[Y_i \leq Y_j\right]}$$

Similarly to the AUC or other ranking metrics, $C = 1$ corresponds to the best model prediction : it is achievable if and only if

$$\forall i, j \quad (\delta_i = 1, Y_i \leq Y_j) \implies f(x_i) \leq f(x_j)$$

which is the case if all predicted survival times are ordered like the life durations, and $C = 0.5$ represents a random prediction.

Here note that if the survival times are, for example, $(T_i = 5, T_j = 6)$, and our model predicts $(\hat{T}_i = 1, \hat{T}_j = 24)$, they are in the right order and the scoring will be the same as with having predicted the actual targets $(\hat{T}_i = 5, \hat{T}_j = 6)$, even though the predictions are quite far from the real values, whereas if we predict $(\hat{T}_i = 5.51, \hat{T}_j = 5.49)$ our predictions are much closer to the ground truth but they are not in the right order so it will be considered as a false prediction.

Transposition to the risk score

In their 2018 paper *RNN-SURV: a Deep Recurrent Model for Survival Analysis*, Giunchiglia, Nemchenko and Van Der Schaar propose using the concordance index as an optimization metric for a deep recurrent neural network. To that end, they compute a risk score $\hat{r}_i$ for all individuals within a batch and then evaluate their ranking using the C-index.

Definition 4.4.1. In the case of a predicted risk score $\hat{r}$ and not a survival time $\hat{T}$, the C-index becomes :

$$C = \frac{\sum_{i,j} \delta_i \mathbb{1}\left[Y_i \leq Y_j\right] \mathbb{1}\left[f(x_i) \geq f(x_j)\right]}{\sum_{i,j} \delta_i \mathbb{1}\left[Y_i \leq Y_j\right]}$$

Adaptation to a loss function

The obvious issue with the concordance index is that, as it is a counting metric (assessing how many pairs of individuals are in the right order), it does not satisfy the main requirement we have for a loss function, namely to be nicely derivable. Therefore, we will transform the C-index in a loss that will have the same ranking properties but that will enable gradient descent.

In their 2007 paper *On Ranking in Survival Analysis: Bounds on the Concordance Index*, Raykar, Steck, Krishnapuram, Dehing-Oberije and Lambin introduce a lower bound on the C-index that will be easier to optimize.

We will first try to find a lower bound on the indicator function. To that end, consider the sigmoid function

$$\sigma(z) = \frac{1}{1+e^{-z}} \iff log(\sigma(z)) = -log(1+e^{-z})$$

We have : $\forall z \quad 0 < \sigma(z) < 1$. Therefore : If $z \geq 0$, $\mathbb{1}\left[z \geq 0\right] = 1$ and :

$$0 < e^{-z} \leq 1 \implies 0 < log(1+e^{-z}) \leq log(2) \implies -1 \leq \frac{log(\sigma(z))}{log(2)} < 0$$

$$0 \leq 1 + \frac{log(\sigma(z))}{log(2)} < 1 = \mathbb{1}\left[z \geq 0\right]$$

If $z < 0$, $\mathbb{1}\left[z \geq 0\right] = 0$ and :

$$1 < e^{-z} \implies log(2) < log(1+e^{-z}) \implies \frac{log(\sigma(z))}{log(2)} < -1$$

$$1 + \frac{log(\sigma(z))}{log(2)} < 0 = \mathbb{1}\left[z \geq 0\right]$$

We conclude :

$$\forall z, \quad 1 + \frac{log(\sigma(z))}{log(2)} < \mathbb{1}\left[z \geq 0\right]$$

In definition 4.4.1 of the C-index, we can reformulate

$$\mathbb{1}\left[f(x_i) \geq f(x_j)\right] = \mathbb{1}\left[z \geq 0\right] \text{ with } z = f(x_i) - f(x_j)$$

As everything is positive, we deduce a lower-bound on the C-index :

$$C > \frac{1}{\sum_{i,j} \delta_i \mathbb{1}\left[Y_i \leq Y_j\right]} \sum_{i,j} \delta_i \mathbb{1}\left[Y_i \leq Y_j\right] \left(1 + \frac{log(\sigma(f(x_i) - f(x_j)))}{log(2)}\right)$$

And finally, the authors define their loss $\mathscr{L}_2$:

$$\begin{aligned}\mathscr{L}_2 &= -\frac{1}{\sum_{i,j}\delta_i \mathbb{1}\left[Y_i \leq Y_j\right]}\sum_{i,j}\delta_i \mathbb{1}\left[Y_i \leq Y_j\right]\left(1 + \frac{log(\sigma(f(x_i) - f(x_j)))}{log(2)}\right) \\ &= -\frac{1}{|\mathscr{E}_{i,j}|}\sum_{\mathscr{E}_{i,j}}\left(1 + \frac{log(\sigma(f(x_i) - f(x_j)))}{log(2)}\right)\end{aligned}$$

You can see from what precedes that it is computationally equivalent (in cost) to compute the exact C-index or its lower bound, which becomes an upper bound on the loss $\mathscr{L}_2$.

The interest of using the lower bound instead of the explicit formula for the C-index lies in the fact that you can't compute a gradient for an indicator function, whereas the sigmoid function is nicely derivable, which will allow us to optimize our model with respect to the C-index, therefore optimizing its ranking capacity. By optimizing the upper bound on the $\mathscr{L}_2$ loss, we guarantee that the loss is decreasing towards its minimum value of -1, even if we don't have the explicit C-index it is sufficient at this stage.

The second formulation of $\mathscr{L}_2$ can seem much more natural to understand, but the reason why I elaborated on the first formulation is that I implemented the loss by the Hadamard product of the

$$\left(1 + \frac{log(\sigma(f(x_i) - f(x_j)))}{log(2)}\right)_{i,j}$$

matrix with a mask, corresponding to the

$$(\delta_i \mathbb{1}\left[Y_i \leq Y_j\right])_{i,j}$$

matrix. This the same methodology that will be used for $\mathscr{L}_1$, see next section.

Comments

We have seen previously that the sign of $1 + \frac{log(\sigma(z))}{log(2)}$ is the sign of z. Therefore, you can notice that the model is able to "compensate" positive loss from misranked individuals with negative loss from correctly classified individuals, therefore optimizing the lower bound on the C-index (upper bound on the loss) would not necessarily guarantee a good ranking fit. However, note that the negative values cannot reach lower than -1, whereas the positive values can theoretically skyrocket to $+\infty$: this limits the compensation ability of the model.

It seems obvious, at first sight, that the $\mathscr{L}_2$ loss is highly dependent on the batch size : the bigger the batch, the more individuals we have to compare, and probably the better the final, optimized ranking will be. This will be an issue at training time, since we will be limited in the maximal batch size acceptable by the machine's memory depending on the model size.

One has to wonder what values this $\mathscr{L}_2$ loss would take in case this sample is empty, that is to say, if we have a batch where all individuals are censored (or empty). The $\mathscr{L}_2$ loss varies from -1 (perfect ranking) to $+\infty$; we decided to put the default value in that case at 0, for several reasons :

- First, it corresponds to the value in the case where all the predicted risk scores are equal :
  $$\sigma(0) = 1/2, \quad 1 + \frac{log(sigma(0))}{log(2)} = 0$$
  Therefore it corresponds to the case where the model has no information about the rankings of the individuals ; in the case with no individuals, we also have no information.
- Second, it enables us to have a more coherent, smoother definition of the loss than is stated in the paper : instead of filtering individuals in the sum (which is costly), we just apply a mask to the matrix of predictions, which is much more efficient. In the case with no individuals, we just mask all the matrix.
- Third and finally, it is coherent with the expression of a sum.

4.4.3 Censored Cross-Entropy Loss

Optimizing the upper bound on the C-index is not a sufficient optimization method, since it has no direct effect on the actual predictions (hazard rate, survival times, etc).

Therefore, Giunchiglia et al. introduce a second loss metric $\mathscr{L}_1$, which is a censored cross-entropy loss (refer to the presentation of the binary cross-entropy loss for further details), to account for the accuracy of the prediction of the survival probability at each time step k :

$$y_k = \mathbb{P}(T > t_k | T > t_{k-1}) = \mathbb{P}(T > t_k | T \geq t_k), \quad y_0 = \mathbb{P}(T > t_0)$$

(the conditional probability doesn't make sense at time step 0 since the individual is necessarily alive at entry in the portfolio).

$$\begin{aligned}\mathscr{L}_1 &= -\sum_{k=1}^{K}\sum_{i\in U_k}\Big(\mathbb{1}\left[Y_i > t_k\right] log(\hat{y}_i^{(k)}) + \\ &\qquad (1-\mathbb{1}\left[Y_i > t_k\right]) log(1-\hat{y}_i^{(k)}\big) \\ &= -\sum_{k=1}^{K}\sum_{i}(\delta_i + (1-\delta_i)\mathbb{1}\left[Y_i > t_k\right])(\mathbb{1}\left[Y_i > t_k\right] log(\hat{y}_i) + \\ &\qquad (1-\mathbb{1}\left[Y_i > t_k\right]) log(1-\hat{y}_i))\end{aligned}$$

With $U_k = \{i|\delta_i = 1 \vee C_i > t_k\}$, the set of individuals who, at time t_k, are *uncensored*, meaning that either *they will be censored later or they will never be*, no matter whether they are *dead or alive* at the moment.

Equivalently speaking, it is the set of individuals for whom we do know, at time t_k, that they are *either dead or alive* (that makes sense since, in our model, we have a third possible state, *censored*, where we don't know if the individual is dead or alive).

This is a standard cross-entropy loss, except that it is calculated on the individuals that either have complete observations, no matter whether they have died yet or not, or will be censored but are not yet. To sum things up, only complete informations up to this point are taken into account.

Similarly to $\mathscr{L}_2$, one has to wonder what values the $\mathscr{L}_1$ loss would take in the case where the sample is empty. The default value is 0, which corresponds to the case where the model is perfect ($\hat{y} = y$), it does make sense since in a case where you have no information at all, there is nothing to optimize. It also makes sense when the expression is a sum.

The use of a binary cross-entropy loss can make one think of a binary classification problem. Indeed, with respect to the $\mathscr{L}_1$ loss, the problem we optimize is K classification problems of knowing whether the individual is still alive or not at each time step.

This loss is computed using a mask matrix, similarly to the C-index, where the mask matrix is multiplied with the uncensored loss with a Hadamard product, which accounts for the U_k subset.

4.4.4 Optimization

The 2 previous losses are linearly combined into a final loss which is being optimized during the training:

$$\mathscr{L} = \alpha\mathscr{L}_1 + \beta\mathscr{L}_2$$

It is equivalent, by a multiplying factor, to

$$\mathscr{L} = \mathscr{L}_1 + \beta \mathscr{L}_2$$

In machine learning in general, and in deep learning in particular, one must always be very cautious to avoid over-fitting. In our case, a first measure of control is to ensure our model doesn't have more parameters than there are data points in the dataset we are training on ; there are also other features in the structure of the model and the optimization strategy which we explain in the following.

L2 Penalization

We optimize the linear combination of the losses using the Adam optimizer, with an l2 regularization (also called weight-decay), which penalizes the loss by an additional factor $\beta||\theta||_2^2$ where θ is the matrix of weights of the model and β an arbitrary parameter settled at 10^{-5} (which is a common default value for the weight decay). This helps ensure that the model will not overfit by attributing arbitrarily high values to some of the weights.

Dropout

Dropout is a very common regularization method in deep learning, introduced by Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov in the 2014 paper *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*. It consists, during the course of the training, of randomly setting the outputs of some nodes to 0, so that to force the model not to attribute too much weight to one node in particular, thus preventing overfitting. The nodes are randomly selected on a Bernoulli law of probability p, and set to 0 ; the others are multiplied by a $\frac{1}{1-p} > 1$ factor, so as to preserve the mean value of the nodes and avoid mechanically vanishing the gradients.

The overall process of dropout is illustrated in figure 4.16. Of course, this random setting of a fraction p of the weights to zero is changing at each time step, so that it averagely hits all weights during the training process, and isn't applied in the validation and testing phase, where we are not optimizing anymore but evaluating the quality of our model on unseen data. We have seen previously that the effects of dropout and first momentum regularization of the Adam optimizer compensate each other ; thus, I have fixed the values for the first and second momentum at their standard default values ($\beta_1 = 0.9, \beta_2 = 0.999$), and I have selected the dropout rate using a cross-validation scheme.

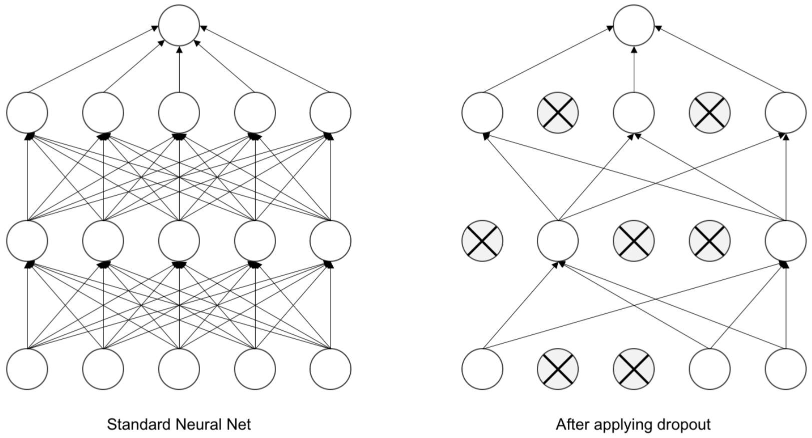


Figure 4.16: The mechanism of Dropout during the training phase

Learning rate scheduler

The proper setting of the learning rate is essential for learning. Indeed, if the learning rate is too big, one might miss the global minimum ; it it is too small, one might fall in a local minimum and not find the global minimum. Therefore, a number of different learning rate schedulers exist, enabling to adapt the learning rate during training : warm-up for starting at a mid-value, increasing learning rate once the slope of the gradient has been found ; cosine scheduler, for alternating between high and low values during training ; and several decrease schedulers (exponential, linear, ...) to gain precision at the end of learning.

My training has been set to start with a learning rate of 10^{-2}, and is scheduled on an exponential decrease :

$$lr(epoch) = 10^{-2} \times 0.995^{epoch}$$

As my training runs for 50 epochs, the learning rate ends on a value of ≈ 0.008.

Gradient clipping

I choose to present here the gradient clipping method that I implemented. The study of gradient values (see figure 4.17) shows very clearly that we have a vanishing gradient problem, thus harming training. It is therefore necessary to ensure gradients stay within a specified range of amplitude, so that the learning rate actually enables us to decide on the size of the training step we take descending the gradient.

The first idea was to add gradient clipping to the Adam with weight decay optimizer we are using. Gradient clipping should indeed help avoid vanishing or exploding gradients. Exploding gradients can occur when the gradient becomes too large and error gradients accumulate, resulting in an unstable network ; on the contrary, due to the use of the sigmoid activation function, vanishing gradients occur when the norm of the gradients

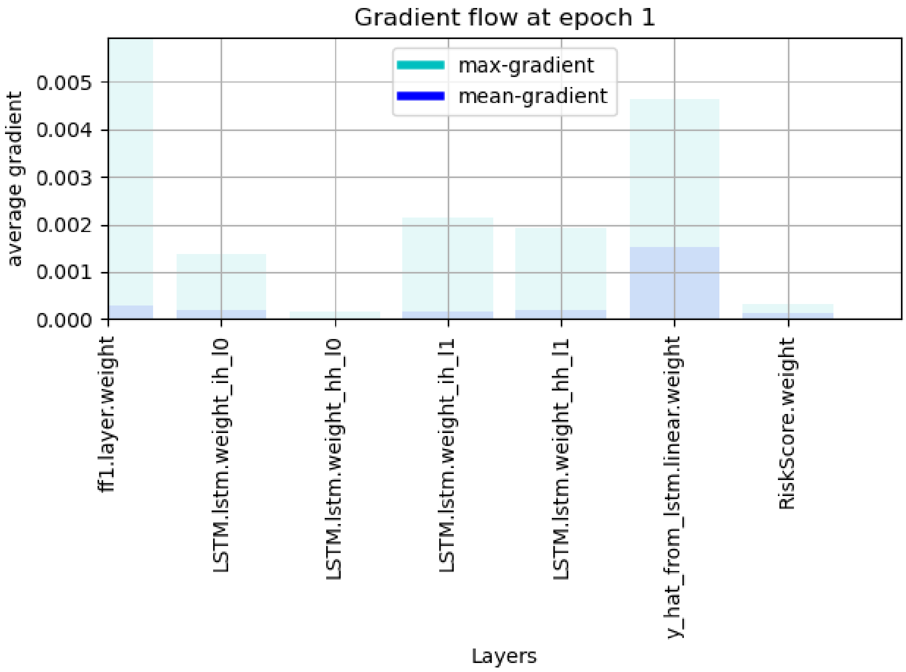


Figure 4.17: Gradient Flow at Epoch 1, No Clipping

tends towards zero, particularly in recurrent networks, and thus the model doesn't learn anymore. Gradient clipping prevents this issue in the gradients that mess up the parameters during training.

There are two possible kinds of clipping :

1. Norm clipping to a value λ :

$$clipped(g_t) = min(\lambda, ||g_t||).\frac{g_t}{||g_t||}$$

   It is not fundamentally very different from just applying a lower learning-rate, but not strictly equivalent, since it softens the regularizing impact of the first momentum in the Adam optimizer.

2. Value clipping to λ : for each coordinate of g_t, you clip it into the interval $[-\lambda, +\lambda]$. The sign of each coordinate is unchanged, you just cannot move too much in any direction.

Each method has its specificity. Namely, norm clipping is good for preserving the overall direction of the gradients ; however, if one component is huge comparatively to the others, normalizing will have the effect of crushing the other components to zero, thus not optimizing any more in these directions. This bias is avoided with value clipping, where smaller gradient values are preserved, however the overall direction of the gradient vector is modified.

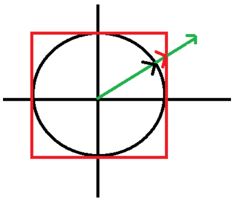


Figure 4.18: Illustration of the clipping by norm and value

An instinctive representation of that nuance, as in figure 4.18 in 2D, is imagining your gradient space as a plane. Norm clipping is equivalent to restricting your gradients to the circle centered on 0 of radius λ, and value clipping is equivalent to restricting your gradients to the square whose corners are $(\pm\lambda, \pm\lambda)$.

From that point, by looking at the gradients values, I noticed that starting from clipping values of 10^{-3} or 10^{-4}, nearly 100% of all gradients coordinated were being clipped. That is to say, they were replaced by their sign times the clipping value. This is what lead me to implementing a sign-Adam : why not just take the sign of each gradient coordinates, still allowing zeros in case of null coordinates ? Here again, it puts all gradient values to ±1 so you have to rescale the learning-rate, this time by dividing it by 10. It didn't work as well as expected. This is because the rescaling with the second moment actually made the two methods unequivalent.

The final clipping method I adopted is a mix between value and norm clipping:

1. For all matrices of weights (there is one by layer), divide the gradients by their mean absolute value so that the mean absolute value of gradients is 1;
2. Clip gradient values to ±2.

The order of steps 1 and 2 is important (see figures 4.19 and 4.20). Indeed, due to the recursive nature of the model and the sigmoid activation functions, we mechanically induce a vanishing gradient problem, as is very clear on figure 4.17. Therefore, clipping gradient absolute values to 2 will ensure we have a very small average absolute value, and dividing all gradients by this value will make the biggest gradients explode, as seen in figure 4.20. This is why dividing by the average value first, to amplify vanished gradients, and then clipping the exploding values to 2 is a much more efficient method to ensure we keep the gradients within a suitable range, see figure 4.19.

We end up with a gradient which is essentially determined by its sign, with a controlled range of amplitude, enabling full use of the advantages provided by the learning rate scheduler (it is harder to actually schedule the size of steps through the learning rate when the gradients have a high variability).

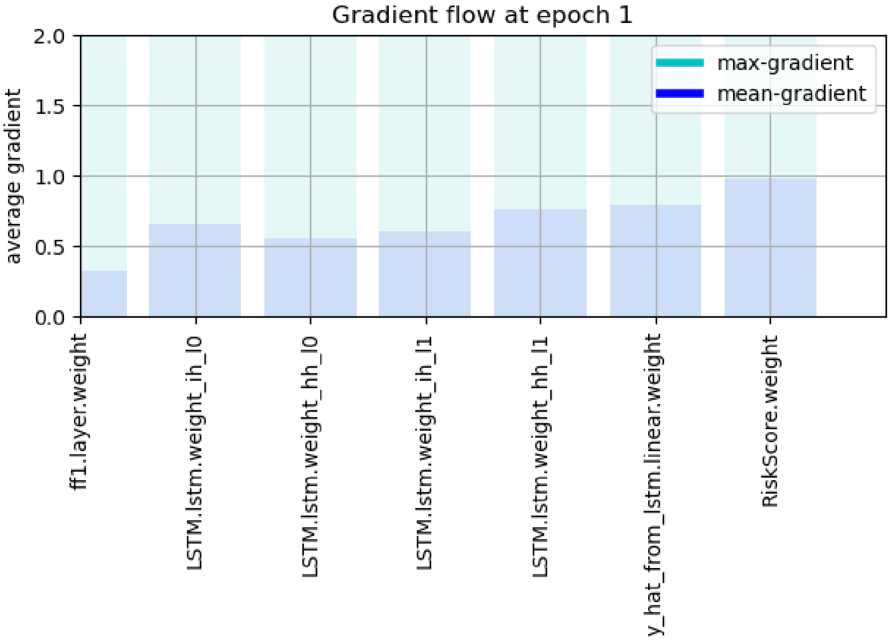


Figure 4.19: Normalization before value clipping

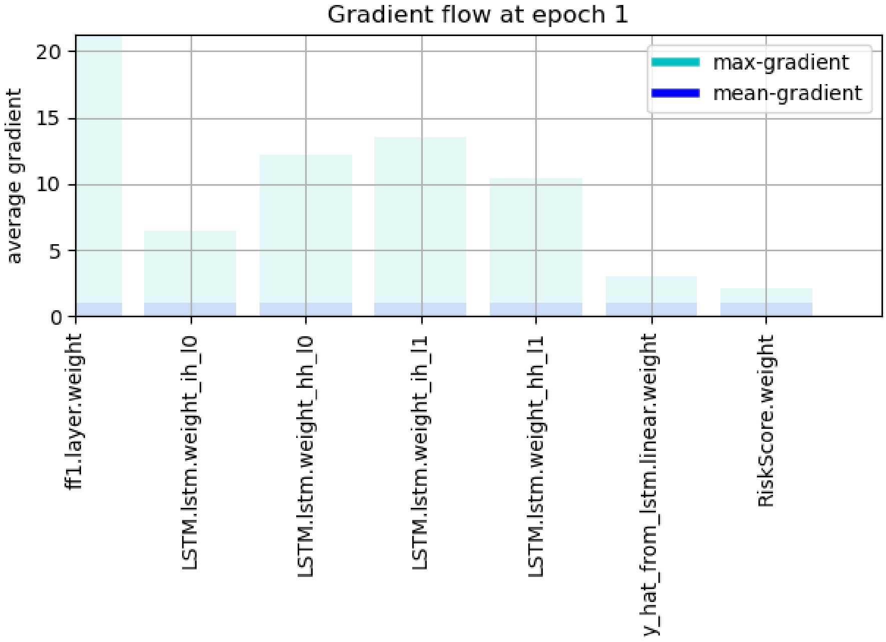


Figure 4.20: Value clipping before normalization

4.4.5 Hyper-parameters cross-validation

Several hyperparameters are optimized through a cross-validation scheme :

- The losses $\mathscr{L}_1$ and $\mathscr{L}_2$ are linearly combined to a loss

  $$\mathscr{L} = \alpha\mathscr{L}_1 + \beta\mathscr{L}_2$$

  I pose $\alpha = 1$ and optimize β through a cross-validation scheme, it is the importance given to $\mathscr{L}_2$ with respect to $\mathscr{L}_1$;
- The architecture of the model : number of embeddings / LSTM layers, hidden sizes of each ; We also tried altogether dropping either the embeddings layer or the LSTM;
- The sharing or not of the $\hat{y}$ head, a single neuron having as entry the hidden size of the LSTM and a bias, and a single output, which is used to convert the output vector of the LSTM at each time step to a scalar, the predicted survival probability $\hat{y}_k$.

All these choices have been considered with respect to the total number of parameters of the model : we must be very careful not to over-parametrize, thus inducing overfitting.

Chapter 5

Pricing applications and scenarii

Measuring the impact of the dynamic evolution of insureds' health features can be used in different pricing schemes.

First of all, dynamic survival models allow actuaries to estimate the sensitivity of a calculated premium to health scenarios. For instance, a possible simulation could estimate the savings related to a person shifting from a high to a normal BMI.

Furthermore, dynamic survival modeling is a method useful for designing coverage with a dynamic pricing. An insured with a healthy or unhealthy lifestyle would receive either discounts or penalties. In this mindset, SCOR developed a Biological Age Model used to price similar contracts.

5.1 Assumptions

As a reminder, the mortality models used in this pricing exercise have been trained on the simulated survival dataset described in a previous section. The machine learning models produce estimations of the mortality rates at each timestep based on the current age, gender, smoker status, BMI, systolic blood pressure and diastolic blood pressure.

In this section, we are interested in pricing a temporary 10-year life insurance policy with a $\$1,000,000$ coverage. The insured is a non-smoker woman aged 45. At the time she applies for the policy, her biometric features (BMI, systolic and diastolic blood pressures) are those of the average woman *in the test database* with the same characteristics:

- $BMI \approx 27.45$
- $BP_{Systolic} \approx 112.5$
- $BP_{Diastolic} \approx 76.5$

This hypothetical woman is slightly overweight but not obese ($BMI \in [25, 30]$), with optimal systolic and diastolic blood pressures (see appendix 5.5) ; therefore, we can consider that she is in an ideal health condition with respect to the parameters.

5.2 Models Training and Testing

The total 200,000 individuals in the generated database were randomly split as follows into 3 subdatabases :

- 60% (120,000 individuals) in a *train* database ;
- 20% (40,000 individuals) in a *validation* database ;
- 20% (40,000 individuals) in a *test* database.

All classic machine learning models and my RNN-Surv implementation have been trained on the train database, with improvements in learning evaluated at each step on performance on the validation set. This is a safety measure to prevent the models from overfitting the train data too much (learning "by heart" some behaviors). Finally, they have been estimated for the pricing simulation on the test database, never seen before, in order to measure the robustness of the inference that have been learned.

5.2.1 Machine Learning Models

The general setup for training machine learning models is a static framework : at each time step t, for each input of the present vector of covariates x_t, we predict a binary target $0/1$ indicating whether the individual survives at $t + 1$.

I will report the results for the following machine learning models from the SCOR survival library :

- Gradient Boosting methods : LightGBM, XGBoost, CatBoost ;
- Poisson and Logistic Regression.

Generalized Linear Models, Generalized Additive Models and Random Forests performances were too poor and were soon abandoned.

The importance attributed by each of these models to the different input covariates are illustrated in figure 5.1. It is normal to have the age as the most important feature, that is coherent with what we observe in the mortality tables (exponential increase in the mortality rate with age). Having gender in second position is quite natural as well. Interestingly enough, these 2 features are not explicitly taken into account in the risk

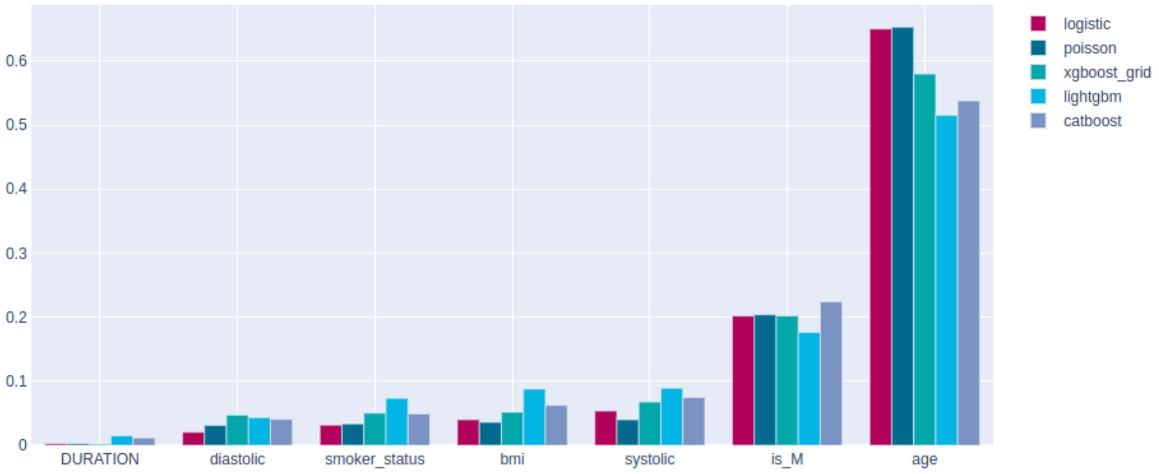


Figure 5.1: Feature importance for the machine learning models

function generating hazard rates, but only implicitly as they influence the underlying multiplier rate extracted from the INSEE mortality tables.

However, you can notice that BMI is deemed less important than systolic blood pressure by most models. First, that is certainly not very signficant. Second, that is due to the fact that these machine learning models are static and incapable to derive risk from past evolution of the features : as past evolution of the BMI is an important risk driver in my hazard rate generation, is is logical that with RNN-Surv I observe an inversed and much more pronounced order. Finally, the low position of diastolic is due to the fact that it doesn't intrinsically add much more information than systolic, being correlated at more than 97%.

5.2.2 RNN-Surv

Interpolation and Padding

The choice I made was to add an is_PAD feature as the last covariate, which is 1 if the time step is after the end date and 0 otherwise. Then, for the default padding vector, I had to choose default values ; I made the arbitrary choice to set a vector of 0, since various testing with other vectors didn't show any significant difference in final performance.

During the simulations, there was little to no difference between padding with a specific padding vector (namely a vector of 0) or forward filling the last observed value. In the case of forward filling the last observed value, one could drop one category for categorical variables ; this made little difference as well, only reducing the size (number of parameters) of the model.

However, what did make an important difference in the final loss values and in the quality of the survival predictions was the introduction of the is_PAD feature as the last covariate. Since it was not mentioned in the original paper, I had supposed that the masking of the losses would be sufficient for the optimizing process, but in the end it appeared crucial to explicitly set the padding indicator.

Framework

As the matrix C caching uncomparable individuals for the loss $\mathscr{L}$ is of size $n \times n$, with n the number of individuals, memory issues mean that it is impossible to optimize all the train database in one time. Therefore, we split it in 12 batches of size 10,000 each which are shuffled between each of the 100 epochs so that it remains as representative as possible.

An extensive hyperparameter optimization grid-search has lead to defining the following setup for our model :

- No embeddings layer : this is a rather radical choice since that brings the model back to a simple LSTM. However, as the primary function of an embedding layer is to get a mapping to a lower dimensional space, one must wonder whether that is pertinent with only 6 covariates + the padding feature, in total an input dimension of only 7. It rather leads to an overparametrization of the problem. The raw features are directly fed as input to the LSTM.
- A LSTM with 2 layers and a hidden size of 5 ;
- α and β in proportions of 10 to 1, for example $\alpha = 5, \beta = 0.5$;
- a dropout rate of 10% ;
- a batch-size of 10,000 ;
- 100 epochs training with an initial learning rate of 10^{-2}, scheduled in 0.95^{epoch}
- sharing the output node of the LSTM to the survival probabilities estimates.

Using the conditional survival probability $y_t = \mathbb{P}(T > t | T \geq t)$ at each time step as target has yielded much better results on all metrics than using the hazard rate $h_t = \mathbb{P}(T = t | T \geq t)$, as was proposed in *Deep Recurrent Survival Analysis* (Ren et al, 2018).

For reproducibility purposes, it is essential to initialize all random generators' seeds for both the CPU and GPU, as they determine both the initial values of the model's weights and the batch order in which individuals from the train database will be sampled. For quality optimization of the $\mathscr{L}$ loss, it is also essential to shuffle batches between each epoch, so that individuals are statistically likely to meet all the others at one point and the comparison loss can be as representative as possible.

5.3 Models Evaluation

5.3.1 Scenario Formulation

It is interesting to mention here that, from a central health scenario with nearly constant BMI and blood pressure, I derived an increasing BMI and a decreasing BMI scenario following 2 methodologies :

- Adding a trend only to the BMI ;
- Adding a trend to the BMI, systolic and diastolic blood pressures.

You can remember from table 3.12 that the BMI was correlated at respectively 34% and 28% with the systolic and diastolic blood pressures, and from the machine learning models presentation that one central assumption was the independence of the covariates, which is clearly not the case here.

In the end, RNN-Surv alone was able to provide updated mortality predictions for a person with single BMI augmentation or decrease, all classic machine learning models kept outputting the same predictions and weren't able to take into account the change in BMI trend, probably because of a confusion with the untouched blood pressure signal. It could also be linked to the fact that they are in a static setup and have no notion of the actual evolution, contrarily to RNN-Surv. However, when all 3 features follow a similar evolution, machine learning models all produce interesting updated mortality estimates.

5.3.2 Payments

For a life insurance of nominal 1,000,000, at each time step $t \in [0, 9]$, if we set the actualisation rate at $IR = 2\%$ the present value of the nominal is

$$PV = \frac{1}{(1 + IR)^{t+1}} N$$

It gives us the following actualisation rates and present values of the nominal at each time step :

```
t       déflateur      cash_pv
0       0.9804         980392
```

```
1         0.9612          961169
2         0.9423          942322
3         0.9238          923845
4         0.9057          905731
5         0.8880          887971
6         0.8706          870560
7         0.8535          853490
8         0.8368          836755
9         0.8203          820348
```

As a reminder, the models output for each time step t_k an estimate of the survival probability at this time step $\hat{y}_k = \mathbb{P}(T > t_k | T > t_{k-1})$, $\hat{y}_0 = \mathbb{P}(T > t_0)$ (the conditional probability doesn't make sense at time step 0 since the individual is necessarily alive at entry in the portfolio). Therefore, based on conditional probabilities, the probability of death at timestep t_k is

$$\mathbb{P}(T = t_k) = \mathbb{P}(T = t_k | T > t_{k-1})\mathbb{P}(T > t_{k-1}) = [1 - \mathbb{P}(T > t_k | T > t_{k-1})]\mathbb{P}(T > t_{k-1})$$

It is easy to see iteratively with conditional probabilities that

$$\begin{aligned}
\mathbb{P}(T > t_0) &= \hat{y}_0, \\
\mathbb{P}(T > t_1) &= \mathbb{P}(T > t_1 | T > t_0)\mathbb{P}(T > t_0) = \hat{y}_1 \times \hat{y}_0, \\
\mathbb{P}(T > t_2) &= \mathbb{P}(T > t_2 | T > t_1)\mathbb{P}(T > t_1) = \hat{y}_2 \times \hat{y}_1 \times \hat{y}_0, \\
\mathbb{P}(T > t_i) &= \prod_{j=0}^{i} \hat{y}_j
\end{aligned} \tag{5.1}$$

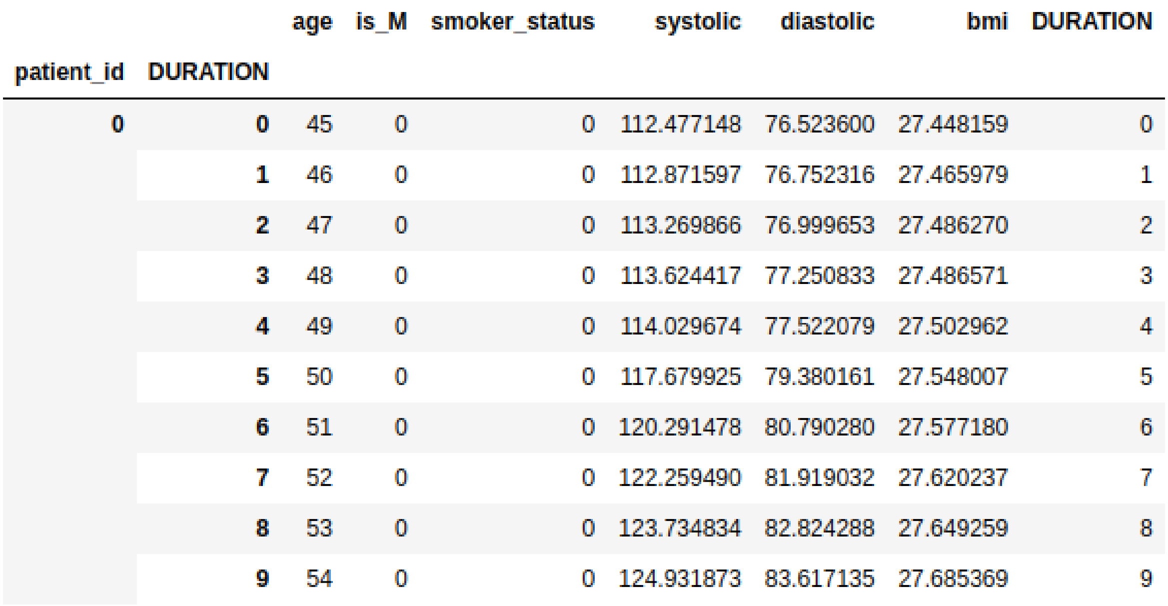
Therefore, the estimated premiums for each model survival estimates are, at each time step k,

$$\frac{N}{(1 + IR)^{t+1}}(1 - \hat{y}_k)\prod_{j<k} \hat{y}_j$$

5.3.3 Central health scenario

The central health scenario is a scenario where the evolution of the BMI and systolic/-diastolic blood pressures during the policy duration follows the average. The table 5.2 represents the central health scenario. Each of these biometric covariates slightly increases as the person gets older over time ; this is very natural and due to the fact that

1. The initial values of the parameters, as derived from the initial distribution, are generally positively correlated with age ;



patient_id	DURATION	age	is_M	smoker_status	systolic	diastolic	bmi	DURATION
0	0	45	0	0	112.477148	76.523600	27.448159	0
	1	46	0	0	112.871597	76.752316	27.465979	1
	2	47	0	0	113.269866	76.999653	27.486270	2
	3	48	0	0	113.624417	77.250833	27.486571	3
	4	49	0	0	114.029674	77.522079	27.502962	4
	5	50	0	0	117.679925	79.380161	27.548007	5
	6	51	0	0	120.291478	80.790280	27.577180	6
	7	52	0	0	122.259490	81.919032	27.620237	7
	8	53	0	0	123.734834	82.824288	27.649259	8
	9	54	0	0	124.931873	83.617135	27.685369	9

Figure 5.2: Average BMI, systolic and diastolic blood pressures for a 45-year-old non-smoker female over 10 years

2. For the 3 parameters, there are more individuals being generated with upward trends than with stable or downward trends.

3. You can remember (figure 3.12) that we had, in the global database, the following correlation rates :

	BMI	Systolic	Diastolic
age	3.7%	34.7%	38.2%

All in all, this scenario will be considered as central since the evolution involved are not of significant amplitude and rather related to age.

The mortality rates, as predicted at each time step (i.e. sequentially) by the different machine learning models, are presented in figure 5.3.

Now, if we set the actualization rate at 2%, we obtain the annual premiums corresponding to each model's mortality estimations, as presented hereunder. You can notice they can vary considerably depending on the estimation, and that they are consistently underestimated ; in the end, RNN-Surv proves to be the best fit.

catboost	lightgbm	xgboost	logistic	poisson	rnnsurv	BE
1977	1508	1411	1457	1498	1667	1851
1964	1333	1382	1562	1605	1631	1871
1922	1411	1353	1676	1720	1784	1740
1916	1392	1351	1796	1841	1746	1660
2084	1735	1670	1926	1973	1709	2110
2129	1801	1749	2173	2212	1672	1919

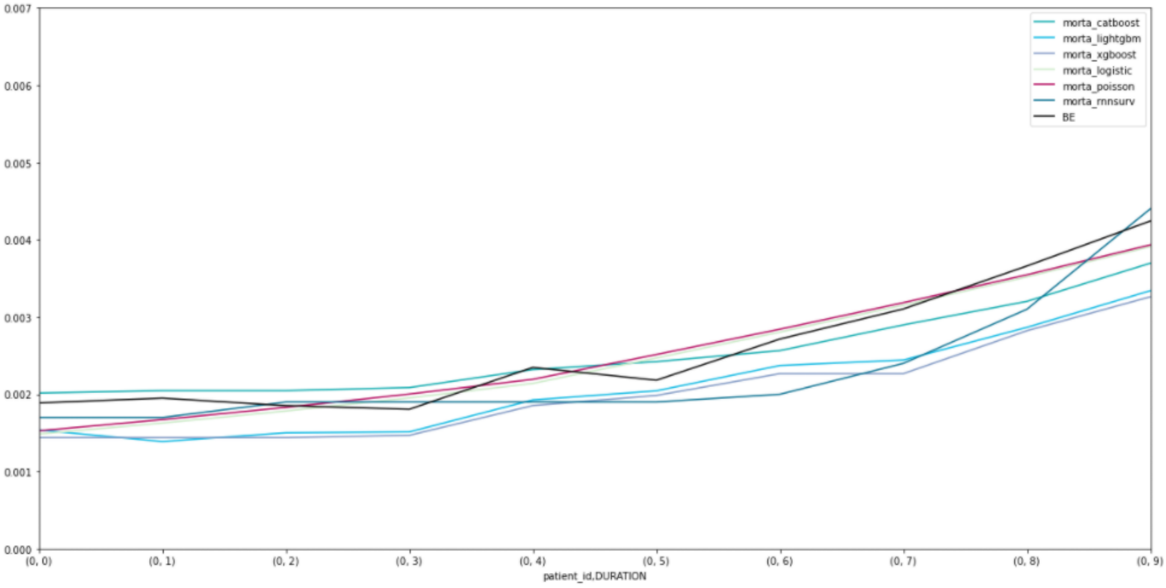


Figure 5.3: Mortality rates of a 45-year-old non-smoker female, central

2204	2045	1955	2413	2445	1722	2335
2435	2060	1912	2653	2680	2022	2611
2631	2363	2328	2893	2915	2554	3008
2967	2692	2631	3142	3159	3543	3406

5.4 Increasing BMI scenario

An increasing BMI scenario reflects a person gaining weight. This can be related to a diet change, certain illnesses, stress, or other factors, as is reflected by the fact that we increase the blood pressure simultaneously. An increasing BMI translates into an increasing coronary heart disease, stroke and cardiovascular death.

To price a sensitivity, an actuary estimates that the BMI of the applicant is linearly increasing with an annual 0.5 increment from the central scenario, while at the same time the blood pressures increase by an annual 2 points. The corresponding evolution is depicted in figure 5.4. You can notice that, from mild overweight and optimal blood pressure, our insured moves to a type I obesity, grade I hypertension health condition. This is obviously reflective of a degradation of her health condition ; we would expect her mortality estimates to increase. Indeed we notice a sharp increase in mortality rates.

The projected cash flows in this scenario are :

catboost	lightgbm	xgboost	logistic	poisson	rnnsurv	BE
1977	1508	1411	1457	1498	1667	1851

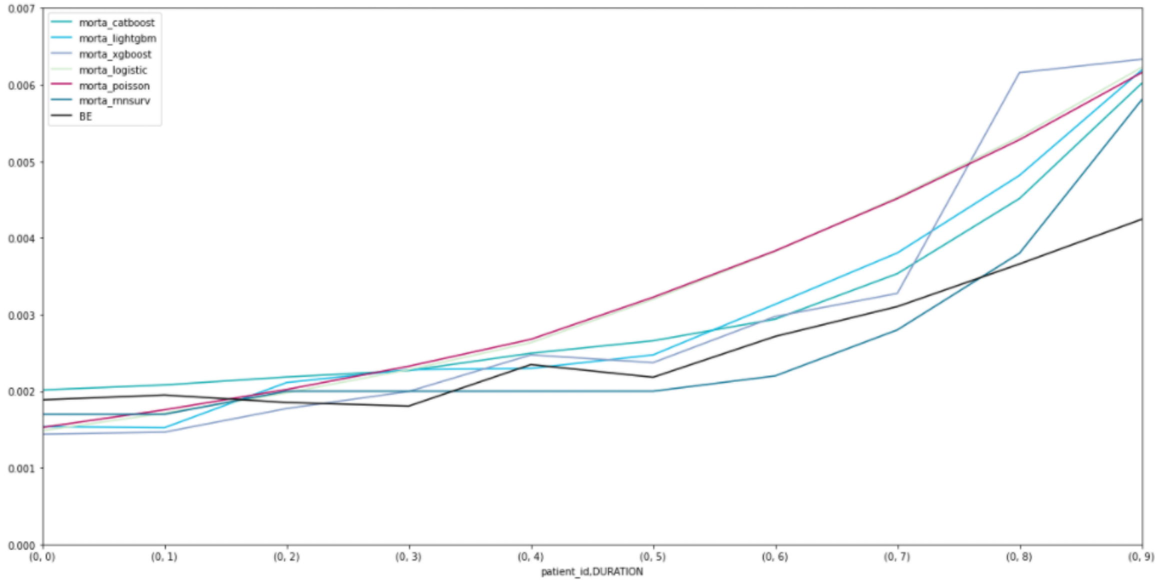


Figure 5.4: Mortality rates of a 45-year-old non-smoker female, ascending

1997	1463	1408	1645	1687	1631	1871
2051	1985	1667	1858	1900	1878	1740
2084	2097	1836	2097	2137	1838	1660
2241	2066	2226	2367	2406	1798	2110
2334	2175	2088	2812	2835	1759	1919
2525	2695	2560	3285	3291	1894	2335
2966	3197	2756	3799	3786	2357	2611
3701	3953	5060	4355	4323	3128	3008
4814	4955	5070	4969	4914	4663	3406

5.5 Decreasing BMI scenario

Contrary to estimating an adversarial scenario for the applicant, an underwriter or a marketing actuary could be interested to estimate how much savings the life insurance company may cash in if the BMI of the insured decrease. Here, the scenario is a decreasing line with annual increment of 0.5 from the central scenario for the BMI, and of 2 points for the blood pressure. The insured falls from overweight to optimal weight, which should be sightly beneficial, but she also falls in hypotension (diastolic). As the model prefers stability, we should expect to see a mortality degradation. The corresponding evolution is depicted in figure 5.5. Indeed we notice a sharp increase in mortality rates.

The projected cash flows in this scenario are :

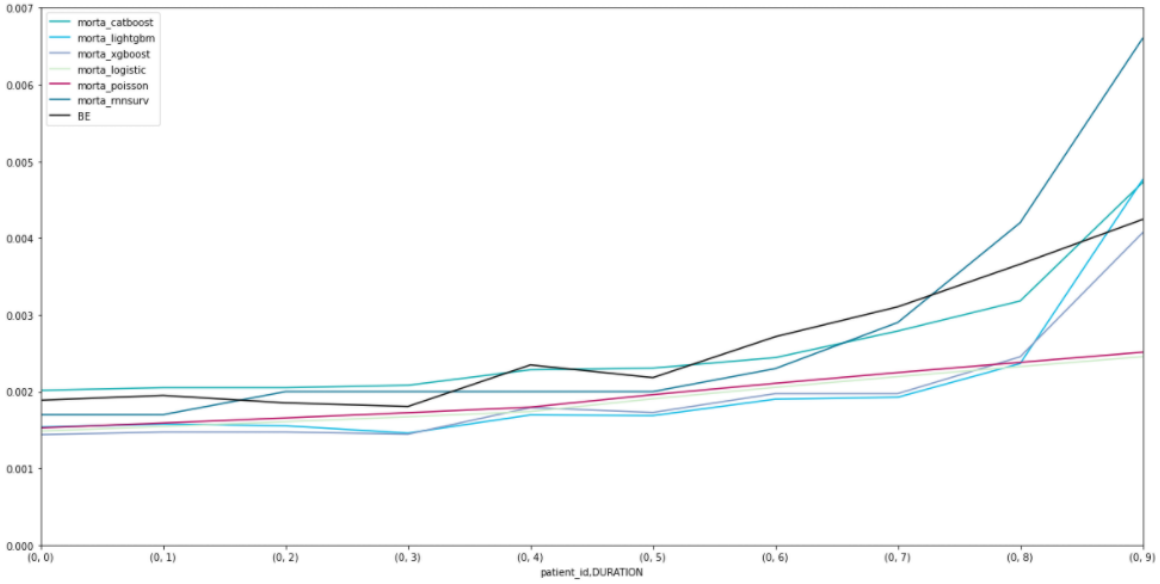


Figure 5.5: Mortality rates of a 45-year-old non-smoker female, descending

catboost	lightgbm	xgboost	logistic	poisson	rnnsurv	BE
1977	1508	1411	1457	1498	1667	1851
1968	1514	1414	1483	1527	1631	1871
1926	1461	1385	1511	1557	1878	1740
1912	1342	1330	1538	1586	1838	1660
2052	1527	1613	1566	1617	1798	2110
2027	1487	1523	1679	1726	1759	1919
2099	1640	1703	1772	1816	1980	2335
2344	1626	1666	1851	1895	2441	2611
2614	1953	2026	1919	1963	3456	3008
3794	3844	3286	1981	2027	5303	3406

The result is very interesting. Indeed, you can notice that the expected cashflows forecast by RNN-Surv are lower than in the increasing scenario for the first 5 years, and then higher : this corresponds to attaining the optimal weight, then a degradation of the blood pressure.

Conclusion

This actuarial thesis aimed to brainstorm on the possibilities of modelling time varying health related covariates in life insurance. These new dynamic mortality models are tools for actuaries to estimate premium variations sensitivities based on the future evolution of the insured biometric profile.

A seriatim dataset was simulated to replicatea life insurance portfolio with realsitic biometric profiles. Insured mortality rates and biometric covariates were evaluated over 26 years and on a annual basis. A censorship mechanism is included in the simulations to create observations with partial information (no event observed) - reccurent in survival data. The correlations, interactions and dynamics have been simplified in order to enable large-scale simulation, although remaining realistic enough of a population. The insured smoking status does not change over time.

The second objective was to examine the sensitivity of machine learning algorithms adapted to model the mortality of dynamic biometric measures : BMI, systolic blood pressure, diastolic blood pressure. These machine learning models are based on the assumption of independence of health measurements between time steps. This hypothesis is not relevant as the processus underlying the evolution health is time dependant. In order to tackle this problem, a recurrent neural networks was studied and implemented in Python. After an in-depth review of the existing adaptations of deep learning architectures for mortality risk assessment, this thesis presents a practical implementation of *RNN-Surv*, a recurrent deep learning model aiming at predicting survival probabilities.

Machine learning mortality models can predict the risk of evolving biometric profiles that are very useful for actuaries for example when leading premium sensitivities of life insurance covers. To tackle the issue of independence between time varying covariates, the RNN-Surv's sequential structure is a good approach. It demonstrated an ability to model very different shapes of survival curves.

Certain work would be of interest to deepen this subject. The simulator could be enhanced with the creation of a bayesian networks based on SCOR expertise and litterature. Other time varying biometric features such as Gamma GT, albumine, LDL, HDL, total cholesterol, and others may be added to the generator. Secondly, the RNN-

Surv necessits further research in order to be fully functionnal. The training of the algorithm is not complete as the average mortality is higher than expected in the dataset. Finally, to confirm the real benefit of this model for mortality modeling, it would be interesting to conduct this experiment on the IMRD database, a 19 million UK patient database of sequential biomedical data, which will enable us to test the performance of our GPU-optimized model on a real world big database.

Data Sources

The following data tables have been used as a support for the initial database distribution of features.

From *Statistique Canada*[1], 2019 prevalence of obesity, hypertension and smokers among the adolescent and adult population of Canada, by age group and gender, in % of the total :

		IMC obese	Hypertension	Fume
12-17	H	28.5	0.4	2.5
	F	20.2	0.5	2.5
18-34	H	35.7	2.6	21.2
	F	24.3	1.6	13.3
35-49	H	43.7	11.7	20.2
	F	31.2	8.2	13.5
50-64	H	41.7	28.2	20.5
	F	33.1	22.5	16.4
65+	H	42.9	42.1	10.0
	F	35.5	44.8	8.4

From *Université de Grenoble*[2], normal and abnormal systolic and diastolic blood pressure range of values :

	Systolic		Diastolic	
	min	max	min	max
Optimal	100	120	70	80
Normal	100	130	70	85
Normal High	130	140	85	90
Isolated Systolic Normal High	130	140	70	85
Grade 1 (light)	140	160	90	100
Grade 1 (light) limit AHT	140	150	90	95
Grade 2 (moderate)	160	180	100	110
Grade 3 (severe)	180	200	110	200

---

[1] https://www.statcan.gc.ca/

[2] http://www-sante.ujf-grenoble.fr

```
Isolated systolic HT              140   200         70    90
Isolated systolic HT limit AHT    140   150         70    90
```

From INSEE[3], 2016 gender ratios (number of males / number of females) by age in the French population :

```
age_min  age_max  ratio
0              4    1.04
5              9    1.05
10            14    1.05
15            19    1.05
20            24    1.03
25            29    1.00
30            34    0.97
35            39    0.96
40            44    0.99
45            49    0.99
50            54    0.99
55            59    0.95
60            64    0.92
65            69    0.89
70            74    0.87
75            79    0.82
80            84    0.69
85            89    0.53
90            94    0.40
95           150    0.27
```

From INSEE[4], 2016 mortality ratios per 10,000 people by age and gender in the French population :

```
         q_F      q_M
age
0       29.04    35.36
1        5.63     6.21
2        2.15     2.39
3        1.33     1.78
4        1.07     1.39
5        0.89     1.18
```

---

[3] https://www.insee.fr/

[4] https://www.insee.fr/

6	0.81	1.08
7	0.74	0.94
8	0.68	0.84
9	0.65	0.79
10	0.66	0.79
11	0.68	0.81
12	0.70	0.86
13	0.79	1.04
14	0.94	1.33
15	1.13	1.71
16	1.26	2.20
17	1.41	2.98
18	1.66	3.93
19	1.92	4.82
20	2.13	5.46
21	2.13	5.87
22	2.14	6.13
23	2.16	6.39
24	2.20	6.61
25	2.31	6.96
26	2.35	7.14
27	2.50	7.37
28	2.66	7.56
29	2.93	7.76
30	3.09	8.01
31	3.28	8.27
32	3.43	8.58
33	3.70	9.10
34	3.89	9.54
35	4.25	10.04
36	4.80	10.54
37	5.45	11.34
38	6.03	12.49
39	6.58	13.62
40	7.25	14.78
41	7.91	16.10
42	8.67	17.47
43	9.92	19.69
44	11.23	21.54
45	12.50	23.61

46	13.81	26.26
47	15.07	29.30
48	17.23	32.11
49	18.17	35.39
50	19.92	38.86
51	21.98	42.35
52	23.44	46.35
53	25.31	51.25
54	27.69	56.99
55	29.63	63.98
56	32.33	69.35
57	34.89	76.68
58	37.42	83.41
59	40.41	88.57
60	42.32	97.89
61	45.60	103.15
62	47.81	110.24
63	51.66	118.14
64	54.01	124.44
65	58.96	131.91
66	61.72	139.02
67	65.89	147.12
68	71.28	158.98
69	77.48	169.28
70	85.96	182.98
71	92.10	195.22
72	102.70	210.19
73	113.47	229.79
74	122.55	246.10
75	137.80	269.25
76	155.27	295.43
77	172.74	328.35
78	192.94	361.18
79	220.11	408.88
80	254.96	453.60
81	294.16	510.26
82	336.77	577.17
83	389.41	646.51
84	451.56	732.59
85	525.00	826.94

86	608.84	936.10
87	706.88	1055.71
88	806.25	1195.32
89	930.69	1331.75
90	1067.83	1481.43
91	1221.44	1655.42
92	1390.13	1838.66
93	1572.35	2031.26
94	1774.97	2252.39
95	1975.88	2474.99
96	2208.76	2669.96
97	2441.61	2883.67
98	2681.44	3139.56
99	2889.20	3384.39
100	3121.83	3447.32
101	3429.02	3593.66
102	3620.43	3820.43
103	3856.58	4056.58
104	3980.77	4180.77

Logrank statistics

The logrank test is the most widely used test to compare survival curves. It is a non-parametric test. This test can be generalized for any number of groups but for the survival tree purpose only the two groups version is used.

In this test, we state the null hypothesis : $H_0 : S_A(t) = S_B(t)\ \forall t$

Within each group we compute the number of expected deaths for each observed time i:

$$e_{Ai} = \frac{n_{Ai} d_i}{n_i} \text{ and } e_{Bi} = \frac{n_{Bi} d_i}{n_i}$$

We finally aggregate for each time to obtain the total number of expected death $E_{.} = \sum_i e_{.i}$ and the observed number of death $O_{.} = \sum_i d_{.i}$.

	Group A	**Group B**	**Total**
Death	d_{Ai}	d_{Bi}	d_i
Survivorship	$n_{Ai} - d_{Ai}$	$n_{Bi} - d_{Bi}$	$n_i - d_i$
Total	n_{Ai}	n_{Bi}	n_i

Table 1: Notation used at time i

Finally the logrank statistics is given by $X^2 = \frac{(O_A - E_A)^2}{E_A} + \frac{(O_B - E_B)^2}{E_B}$

Under H_0 this statistics follows a chi-square distribution with one degree of freedom, we can then compute the p-value.

As the whole survival theory, the validity of this test relies on the independence assumption between the observed event and the censoring.

The main limit is the difficulty to reveal the difference of mortality between two groups when their survival curve cross. The power of the test is indeed maximum for proportional curves.

Bibliography

Aalen, O. (1978). “Nonparametric inference for a family of counting processes”. In: *Annals of Statistics 6, 701–726.*

C Kooperberg, D B Clarkson (1997). “Hazard regression with interval-censored data”. In: *Biometrics.*

Cox, D. R. (1972). “Regression Models and Life Tables”. In: *Journal of the Royal Statistical Society.*

D Faraggi, R Simon (1995). “A neural network model for survival data”. In: *Statistics in Medicine.*

— (1998). “Bayesian variable selection method for censored survival data”. In: *Biometrics.*

Diederik P. Kingma, Jimmy Ba (2015). “Adam: A Method for Stochastic Optimization”. In: *3rd International Conference for Learning Representations, San Diego.*

Eleonora Giunchiglia, Anton Nemchenko and Mihaela van der Schaar (2018). “RNN-SURV: a Deep Recurrent Model for Survival Analysis”. In: *Department of Engineering Science, University of Oxford, UK.*

Hastie, T., R. Tibshirani, and J. Friedman (2008). “The Elements of Statistical Learning”. In: *Springer.*

Hosgood, Giselle Louise (2002). “Markov models to estimate and describe survival time and experience in cohorts with high euthanasia frequency”. In: *Louisiana State University and Agricultural and Mechanical College.*

Kaplan, E. M. and P. Meier (1958). “Nonparametric estimation from incomplete observations”. In: *Journal of the American Statistical Association.*

Kvamme, Håvard (2018). “Time-to-event prediction with PyTorch”. In: *https://github.com/havakv/py*

Laura Löschmann, Daria Smorodina (2020). “Deep Learning for Survival Analysis”. In: *Humboldt Universität.*

Martin, Laurène (2020). “Mortality Risk Modeling with Machine Learning”. In: *Mémoire d’actuariat.*

Nelson, W. (1972). “Theory and applications of hazard plotting for censored failure data”. In: *Technometrics. 14: 945–965.*

Organization, World Health (2020). “Body Mass Index - BMI”. In: *World Health Organization.*

Patrick, Gary (2020). “Reinsurance”. In: *Quant Schools.*

Peter C. Austin, Douglas S. Lee and Jason P. Fine (2016). “Introduction to the Analysis of Survival Data in the Presence of Competing Risks”. In: *Journal of the American Heart Association.*

Rodrìguez, G. (2001). “Non-Parametric Estimation in Survival Models”. In: *Spring.*

Schmid, M., M. N. Wrigh, and A. Ziegler (2016). “On the use of Harrell’s C for clinical risk prediction via random survival forests”. In: *Expert Systems with Applications.*

“The Breadth and Scope of the Global Reinsurance Market and the Critical Role Such Market Plays in Supporting Insurance in the United States” (2014). In: *FEDERAL INSURANCE OFFICE, U.S. DEPARTMENT OF THE TREASURY.*

Tibshirani, R. et al. (2011). “Regularization Paths for Cox’s Proportional Hazards Model via Coordinate Descent”. In: *Journal of Statistical Software.*

Wang, P., L. Yan, and R. K. Chandan (2017). “Machine Learning for Survival Analysis: A Survey”. In: *arXiv:1708.04649.*

Survival_Deep_Learning

Final Audit Report 2022-02-03

Created:	2022-02-03
By:	Louise FERBACH (LFERBACH@scor.com)
Status:	Signed
Transaction ID:	CBJCHBCAABAAmtx9x7AScZp1gN4B3-LercwEvO729qvY

"Survival_Deep_Learning" History

Document created by Louise FERBACH (LFERBACH@scor.com)

2022-02-03 - 8:34:09 AM GMT- IP address: 85.115.60.202

Document emailed to Antoine LY (ALY@scor.com) for signature

2022-02-03 - 8:35:18 AM GMT

Email viewed by Antoine LY (ALY@scor.com)

2022-02-03 - 9:04:36 AM GMT- IP address: 90.83.87.129

Document e-signed by Antoine LY (ALY@scor.com)

Signature Date: 2022-02-03 - 9:04:52 AM GMT - Time Source: server- IP address: 85.115.60.202

Agreement completed.

2022-02-03 - 9:04:52 AM GMT