Survival Analysis: Applications to Credit Risk Default Modelling

By

Mark Thackham

Bachelor of Science(Mathematics), Master of Applied Finance, Master of Applied Statistics, Master of Research(Statistics), Actuary

> A thesis submitted to Macquarie University for the degree of Doctor of Philosophy Department of Mathematics and Statistics 26th February 2021



© Mark Thackham (41411889)

Bachelor of Science(Mathematics), Master of Applied Finance, Master of Applied Statistics, Master of Research(Statistics), Actuary, 2021.

Typeset in $\operatorname{LAT}_{E} X 2_{\mathcal{E}}$.

Except where acknowledged in the customary manner, the material presented in this thesis is, to the best of my knowledge, original and has not been submitted in whole or part for a degree in any university.

Mark Thackham Bachelor of Science(Mathematics), Master of Applied Finance, Master of Applied Statistics, Master of Research(Statistics), Actuary

Dedication

To Jia (殷佳) and Dorothy (殷然), for bringing me so much happiness, you are truly both the light and loves of my life. Thank-you for your love and support, and for the sacrifices you have made to support me on this journey. May the many years ahead be ours to cherish together.

Acknowledgements

The completion of my Doctorate thesis represents culmination of eight continuous years part-time study achieving three degrees in statistics – Master of Applied Statistics; Master of Research (Statistics); and ultimately now a PhD in Statistics – undertaken while also both working professionally full-time in industry and raising a young family. It is also the capstone to five analytic qualifications over 20 years spanning mathematics, finance, actuarial science and statistics. All these years, especially the last eight, have been highly formative personally and professionally for myself and my family. Eight years ago I started newly-married working professionally in Sydney; welcomed our daughter; accepted a professional secondment to New Zealand; then moved with my family to settle into a new life in London where I continued my career in industry working at two different European firms. None of this journey would have been possible without the support and encouragement of my wife and daughter – thank-you to you both and I hope that I can repay the sacrifices that you have made to support me.

I owe a substantial and massive debt of gratitude to my PhD supervisor, Associate Professor Jun Ma. Jun kindly supervised all of my research at Macquarie University, from my MAppStat project, MRes(Stat) thesis and finally to this PhD thesis. I am the statistician that I am today because of Jun's advice, instruction and encouragement. Thank-you very much Jun.

To my parents Anne and Gary, you are wonderful and caring people who instilled in me from a young age the transformative power of education, encouraged me to achieve my best and provided every opportunity for me to do this. To my sister Jen, you are a towering intellect whose advice has been invaluable and has helped keep me grounded just when I needed. Thank-you to all of you for your love, support and encouragement.

Fundamentally as a statistician and Actuary working in industry I have had the absolute true fortune of working with and for some of the brightest minds, keenest intellects and best communicators from each of whom I have learned so much. There are far too many to list here, but I will forever be indebted to each of you.

Despite these above gratitudes (and the countless weekends and evenings foregone), all work, opinions and details (including any remaining errors or omissions) remain my own, and do not represent the views or opinions of any other individual or institution.

Abstract

Credit-granting institutions lend money to customers, some of which may fail to make contractual repayments (namely principal, interest and fees) thereby defaulting on their obligation. Firms employ quantitative credit risk management techniques to estimate and appropriately control their credit risk, ensuring the firm's risk profile remains within its risk appetite, thus contributing to a safely run firm and stability of the wider economy. Quantitative credit risk management techniques are used to estimate: Probability of Default (PD); Exposure at Default (EAD); and Loss Given Default (LGD). These are inputs to calculate expected loss (EL) (for loan-loss provisions required under international accounting standards (IASB (2014), FASB (2016)), as well unexpected loss (UL) (required by institutions granted regulatory approval under the Basel Accords (BIS, 2006) to use the Advanced Internal Ratings Based (A-IRB) Approach for minimum credit capital).

This thesis focuses on applying survival analysis to quantifying the risk of credit default used for PD. Institutions already use their own internal data and leverage analytical techniques to quantify the risk of credit default, so the refinements in this thesis could further assist firms control their credit risk profile. To be granted regulatory and audit approval, quantitative credit risk models need to have intuitive drivers and functional form. Therefore regression approaches are regularly adopted, and while logistic regression is common (Baesens et al. (2003), Lessmann et al. (2015)), survival models achieve comparable accuracy to logistic regression but provide additional benefits, such as including censored data and estimations over multiple time horizons (Bellotti and Crook (2009), Stepanova and Thomas (2002) and Tong et al. (2012)).

Survival analysis describes studies where subjects are followed in anticipation they encounter an event of interest. Originating with Edmund Halley's life table of human mortality (1693) and its extension by Daniel Bernoulli (1760) demonstrating the increase in human survival if the competing risk of small pox were eliminated as a cause of death, survival analysis spans applications across multiple disciplines, such as biomedical science, industrial life testing (Kalbfleisch and Prentice, 2002) and finance (Lessmann et al., 2015). Regression techniques and method of partial likelihood were introduced by David Cox (1972, 1975), and remain prominent (Hosmer et al., 2008). This model has since been extended, particularly by Crowley and Hu (1977) to cater for time-varying covariates, and by (for example) Sy and Taylor (2000) to cater for mixture-cure models.

This thesis explores over three chapters, via two published papers and one manuscript prepared for publication, computational enhancements to the application of survival analysis, competing risk analysis, and mixture-cure analysis, to estimating the risk of credit default. These enhancements are: (1) joint estimation of regression coefficients and baseline

hazard using constrained maximum likelihood, where the constraint ensures the latter's nonnegativity; (2) calculation of an asymptotic variance-covariance matrix that allows inferences to be drawn for regression estimates; (3) improved accuracy of parameters in certain settings as demonstrated via simulation. Applied to credit risk modelling, the methods in this thesis provide comparably accurate regression parameters to those obtained using partial likelihood but with the added benefit of also returning an estimate of a baseline hazard estimate with relatively low variability along with asymptotic variance estimates for the baseline hazard and all regression parameters. This further information allows clearer resolution of the shape and statistical significance of the underlying baseline hazard for the risk of credit default. For survival analysis and competing risk analysis approaches in this thesis, time-varying covariates are included providing additional flexibility of including into the models covariates whose values change over time.

Contents

De	edicat	ion	iv
Ac	know	ledgements	vi
Al	ostrac	t	vii
1	Intr	oduction	1
	1.1	Motivations - Quantitative Credit Risk Management	1
		1.1.1 Background and Motivations	1
		1.1.2 Expected Loss (EL) and Unexpected Loss (UL)	4
		1.1.3 Credit Risk Components - PD, EAD, LGD	6
		1.1.4 Probability of Default (PD)	6
		1.1.5 Estimating Credit Default using Survival Analysis	7
	1.2	Aims and Contributions	9
		1.2.1 Existing Approaches and Limitations	9
		1.2.2 Summary of Contributions - 2 Publications, 1 Manuscript	10
2	Lite	rature Review	12
	2.1	Background - Survival Data and Survival Analysis	12
		2.1.1 Survival Data	13
		2.1.2 Survival Analysis	14
	2.2	Literature Review - Models	15
		2.2.1 Single-Event Survival Analysis	16
		2.2.1.1 Semi-Parametric Cox Model - Baseline Covariates	17
		2.2.1.2 Semi-Parametric Cox Model - Time Varying Covariates	20
		2.2.1.3 Baseline Hazard Estimation	21
		2.2.1.4 Maximum Penalised Likelihood Estimation	22
		2.2.2 Competing Risk Analysis	24
		2.2.3 Mixture-Cure Analysis	27
		2.2.3.1 Estimation via Expectation-Maximisation	29
		2.2.3.2 Estimation via Other Methods	30
	2.3	Literature Review - Applications to Credit Risk	31
	2.4	Structure of This Thesis	35
2	DAD	ED 1. On Maximum Likelihaad Estimation of the Sami Danamatric Cay Model with Time	
3	Vary	ing Covariates	36
	3.1	Abstract	36
	3.2	Introduction	37
	3.3	The likelihood function	39
	3.4	Constrained optimisation	42
	3.5	Asymptotic Properties	44
	3.6	Results	47
		3.6.1 A simulation study	47
		3.6.2 Application 1 - Stanford Heart Data	53
		3.6.3 Application 2 - Credit Risk Data	55
	3.7	Conclusion	57
	3.8	Appendix: Hessian matrix	58

References

	-	4	n
•			
ч.	,	٩	U.

4	PAPER Semi-P	2: On Maximum Likelihood Estimation of Competing Risks using the Cause-Specific arametric Cox Model with Time-Varying Covariates – an Application to Credit Risk	с 65
	4.1 A	ostract	66
	4.2 In	troduction	66
	4.3 T	ne likelihood function	68
	4.4 R	esults	71
	4.	4.1 Simulation Results	71
	4.	4.2 Application to Credit Risk Data	74
	4.5 C	onclusion	79
Re	ferences	nalan artan Mataial	82
	4.0 50		07
Re	ferences		94
5	PAPER	3: Maximum Likelihood Estimation of the Mixture Cure Semi-Parametric Cox Model -	_
	an App	lication to Credit Risk	96
	5.1 A	ostract	96
	5.2 In	troduction	97
	5.3 M	aximum Likelihood Estimation	100
	5.	3.1 The Likelihood Function	100
	5.	3.2 Gradient Vector and Hessian Matrix	103
	5.4 C	onstrained Optimisation Algorithm	104
	5.5 R	esults	106
	5.	5.1 Simulation Results	106
	5.	5.2 Application to Credit Risk Data	108
	5.6 C	onclusion	112
	5.7 A	ppendix A	113
ъ			
ке	ferences		115
ке 6	ferences Conclu	sion and Discussion	115 120
Re 6 A	ferences Conclu Append	sion and Discussion lix - Implementation and Simulation	115120125
Re 6 A	ferences Conclu Append A.1 M	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC	 115 120 125 126
6 A	ferences Conclu Append A.1 M	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC	 115 120 125 126 126
6 A	ferences Conclu Append A.1 M A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls	 115 120 125 126 126 127
6 A	Conclu Append A.1 M A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A 1 2 1 comph.mlt()	 115 120 125 126 126 127 127
6 A	ferences Conclu Append A.1 M A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 corph_mlt control()	 115 120 125 126 126 127 127 128
6 A	ferences Conclu Append A.1 M A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control()	 115 120 125 126 126 127 127 128 129
Ke 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1	 115 120 125 126 126 127 127 128 129 129 129
Ke 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt()	 115 120 125 126 126 127 127 128 129 129 130
Ke 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt()	 115 120 125 126 126 127 127 128 129 129 130 120
Ke 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt()	 115 120 125 126 126 127 128 129 129 130 130 120
Ke 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.4 residuals.coxph_mlt()	1115 120 125 126 126 127 127 128 129 129 130 130 130
Ke 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.35 predict.coxph_mlt()	1115 120 125 126 126 126 127 127 128 129 129 130 130 130
Ke 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt()	115 120 125 126 126 126 127 127 128 129 129 130 130 130 131
Ke 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt()	1115 1200 1255 1266 1276 1277 1277 1288 1299 1300 1300 1300 1310 1311
Re 6 A	ferences Conclu Append A.1 M A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt()	1115 1200 1255 1266 1276 1277 1288 1299 1300 1300 1300 1310 1311 1311
Re 6 A	Append A.1 M A A A A A A A A A A A Si	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt() I.4 R Code A.1	1115 1200 1255 1266 1276 1277 1288 1299 1300 1300 1300 1310 1311 1311
6 A	Append A.1 M A A A A A A A A A A A A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.7 predict.coxph_mlt() I.1.3 R Code	1115 120 125 126 126 127 127 128 129 130 130 130 130 131 131 131
6 A	Anticological Action of the second se	sion and Discussion Iix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt() 1.3 R Code Multating Survival Data with TVC	1115 120 125 126 126 127 127 128 129 130 130 130 130 131 131 131 151
Ke 6 A	Append A.1 M A A A A A A A A A A A A A A A A A A A	Sion and Discussion Iix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt() 1.4 R Code mulating Survival Data with TVC 2.1 R Code 3.1 Background	1115 120 125 126 126 127 127 128 129 130 130 130 130 131 131 151 154 154
Ke 6 A	Append A.1 M A A A A A A A A A A A A A A A A A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt() A.1.3.7 predict.coxph_mlt() A.1.3.8 twith TVC 2.1 R Code 3.1 Background 3.2 Simulating Survival Time	1115 1200 1255 1266 1266 1277 1277 1288 1299 1300 1300 1300 1310 1311 1311 1511 1514 1544 1555
Ke 6 A	ferences Conclu Append A.1 M A A A A A A A A A A A A A A A A A A A	sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt() A.1.3.7 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt() A.1.3.8 code Mulating Survival Data with TVC 3.1 Background 3.2 Simulating Survival Time A.3.2.1 Analytical Inversion of H(t)	115 120 125 126 126 127 127 128 129 130 130 130 130 131 131 151 151 154 155 157
Ke 6 A	ferences Conclu Append A.1 M A A A A A A A A A A A A A A A A A A A	Sion and Discussion lix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.7 predict.	115 120 125 126 126 127 127 128 129 130 130 130 130 130 131 131 151 151 154 155 157 157
Ke 6 A	ferences Conclu Append A.1 M A A A A A A A A A A A A A A A A A A A	Sion and Discussion Hix - Implementation and Simulation odel Implementation - Cox Model with TVC 1.1 Two Input data.frames 1.2 Fitting a Model and Available Controls A.1.2.1 coxph_mlt() A.1.2.2 coxph_mlt.control() 1.3 Exploring a Fitted Model A.1.3.1 summary.coxph_mlt() A.1.3.2 tidy.coxph_mlt() A.1.3.3 coef.coxph_mlt() A.1.3.4 residuals.coxph_mlt() A.1.3.5 predict.coxph_mlt() A.1.3.6 plot.coxph_mlt() A.1.3.7 predict.coxph_mlt() A.1.3.7 predict.coxph_mlt() 1.4 R Code mulating Survival Data with TVC 3.1 Background 3.2 Simulating Survival Time A.3.2.1 Analytical Inversion of $H(t)$ A.3.2.2 Numerical Inversion of $H(t)$	115 120 125 126 126 127 127 128 129 130 130 130 130 130 131 131 151 151 154 155 157 157

References										177											
	A.5.1	R Code		•••		•	•••	•••		•	 •	·	 •	 •	 •	 •	 ·	•	•	•••	173
A.5	Simula	ting Mix	ture-Cu	re Data							 •							•	, .		173
	A.4.1	R Code									 •							•			165
A.4	Model	Impleme	entation	- Mixtu	re-C	Cure	e M	ode	1.		 •										165

List of Figures

1.1	Loans and Advances for Australia's Seven Largest Lenders (as at August 2020, Source: APRA (2020))	4
1.2 1.3	Relationship Between Expected Loss and Unexpected Loss. Source:BIS (2005) Credit Loss Distribution. Source: Yeh et al. (2005)	5 5 °
1.4	Stynsed Example of the Application of Survival Analysis to Cledit Default.	0
2.1 2.2	Semi-Parametric Cox Model Applied to Credit Default Risk Semi-Parametric Cause-Specific Cox Model with Competing Risks Applied to Credit Default Risk	15 15
2.3 2.4	Semi-Parametric Mixture-Cure Cox Model Applied to Credit Default Risk . Estimated Coefficients from for US Credit Card Data Model	15 33
3.1	Simulation Results for $\hat{\gamma}_1$ (with a censoring proportion of (a) 20%, (b) 80%), comparing maximum likelihood (ML) and partial likelihood (PL) estimation, Exponential Baseline.	50
3.2	Simulation Results for $\hat{\gamma}_1$ (with a censoring proportion of (a) 20%, (b) 80%), comparing maximum likelihood (ML) and partial likelihood (PL) estimation, Weibull Baseline.	51
3.3	Comparison of true and estimated Exponential baseline hazards, together with simultaneous 95% CIs.	52
3.4	Comparison of true and estimated Weibull baseline hazards, together with simultaneous 95% CIs.	53
3.5	Comparison of the baseline hazard estimation from the maximum likelihood (a) and the Breslow (b) methods	55
3.6	Comparison of Baseline Hazard Using: "Breslow + Partial Likelihood (PL)" verses "Maximum Likelihood (ML)" Estimation	57
4.1	Maximum Likelihood (ML) and Partial Likelihood (PL) Simulation Results for $\hat{\gamma}_1$	73
4.2	Comparison of true and Maximum Likelihood estimated baseline hazards,	74
4.3	Comparison Between Partial Likelihood and Maximum Likelihood Estima- tion of Cause-Specific Baseline Hazards and Baseline Cumulative Hazard for Default and Redemption. Shaded Regions are 95% Asymptotic Maximum Likelihood Confidence Intervals.	74
5.1	Comparison of true and Maximum Likelihood estimated baseline hazards, to- gether with simultaneous 95% CIs, using Constrained Maximum Likelihood (ML)	107

5.2	Baseline Hazard and Piecewise 95% Confidence Interval for the Susceptible								
	Population	111							
A.1	Competing Risk Analysis Example of 2 Possible Events	154							

List of Tables

3.1	Example of time-varying covariate data frame	41
3.2	Comparing estimates $\hat{\gamma}_1$, $\hat{\beta}_1$ and $\hat{\beta}_2$ from maximum likelihood (ML) and	
	partial likelihood (PL) methods	49
3.3	Parameter estimates for the Stanford Heart Transplant Data	54
3.4	Description of the 11 baseline and 2 time-varying covariates in the Home	
	Loan dataset	56
3.5	Comparison of Parameter Estimates of the Eleven Baseline and Two Time-	
	Varying Covariates Using Maximum Likelihood and Partial Likelihood Esti-	
	mation	56
4.1	Comparing estimates $\hat{\beta}_{11}$, $\hat{\beta}_{12}$, $\hat{\gamma}_1$, $\hat{\beta}_{21}$, $\hat{\beta}_{22}$, and $\hat{\gamma}_2$, from maximum likelihood	
	(ML) and partial likelihood (PL) methods	73
4.2	Comparison of AUC (with 95% CI), from maximum likelihood (ML) and	10
	partial likelihood (PL) plus Breslow methods	73
4.3	Distribution of Observed and Censored Outcomes	75
4.4	Description of the 14 baseline and 3 time-varying covariates in the Home	
	Loan dataset	76
4.5	Parameter Estimates for Maximum Likelihood and Partial Likelihood for	
	Default	76
4.6	Parameter Estimates for Maximum Likelihood and Partial Likelihood for	
	Redemption	77
4.7	Calibration and Discrimination Results the Models Estimated using Max-	
	imum Likelihood (ML) and Partial Likelihood + Breslow (PL+Bres.) for	
	Default	79
4.8	Example of time-varying covariate data frame	88
5.1	Simulation Results of Parameters Estimates using Expectation Maximisation	
	(EM) and Constrained Maximum Likelihood (ML)	107
5.2	Number of Defaults and Observations in the Credit Risk Dataset	108
5.3	Description of the 11 covariates in the Home Loan dataset	108
5.4	Results of Parameters Estimates using Expectation Maximisation (EM) and	
	Analytical Maximum Likelihood (ML)	110
5.5	Calibration Results for Expectation Maximisation (EM) and Constrained	
	Maximum Likelihood (ML)	112
A.1	An Example Baseline Dataframe	126
A.2	Example Time-Varying Covariate Data Frame	127

1

Introduction

This introduction chapter contains two sections. The first motivates why quantitative credit risk modelling is important to individual institutions, regulators, the economy and society. The second outlines existing approaches and their limitations to quantitative credit risk modelling, and summarises contributions by this thesis addressing these limitations.

1.1 Motivations - Quantitative Credit Risk Management

1.1.1 Background and Motivations

Fundamentally credit-granting institutions engage in three core activities: (1) accepting and safeguarding customer deposits; (2) making payments on behalf of customers; and (3) granting credit to customers. While the primary activity amongst these is granting credit, some customers may fail to repay in a timely manner monies they contractually owe (namely principal, interest and fees), thereby defaulting on their obligation. This exposes the institution to **credit risk**, which Apostolik et al. (2009) defines as: *"the potential loss a bank would*

suffer if a borrower fails to meets its obligations".

The term "credit" is the activity of lending money. The term "risk" is broader with no single definition appropriate in all settings. The OED (2020) defines risk as "exposure to the possibility of loss, injury, or other adverse or unwelcome circumstance". McNeil et al. (2005) provides two definitions specifically for quantitative finance which are: "any event or action that may adversely affect an organization's ability to achieve its objectives and execute its strategies", or alternatively "the quantifiable likelihood of loss or less than expected returns". Jorion (2006) similarly states risk can generally be defined as the "...uncertainty of outcomes when compared to expectation", outlining the word "risk" originates from Latin, through the French "risqué" and the Italian "risco". The sense of "risco" is to cut off like a rock, from the Latin "re-" (back) and "secare" (to cut), demonstrating the sense of peril to sailors of sharp rocks.

Despite this, Crouchy et al. (2006) outline effective management risk does not (necessarily) equate to risk avoidance, identifying the "conflict of risk and reward" where higher economic returns (on average) are expected to be earned from higher risk exposure. Lam (2003) describes three steps for effective risk management: (1) risk identification; (2) risk quantification; and (3) risk control. Lam (2003) continues by stating this allows relevant risks to be clearly assessed, objectivity quantified, and appropriately controlled via one or more of: (1) risk avoidance; (2) risk reduction; (3) risk retention; and/or (4) risk transfer. The aim is that any left-over risk, called "residual risk", is lowered to a tolerable level given the institutions "risk appetite", allowing tailoring of a targeted risk-return profile. This thesis focusses on refining and improving risk quantification methods for a specifically identified risk, the risk of credit default. Credit-granting institutions already actively deploy quantitative methods to quantify the risk of credit default, and could use refinements and improvements in this thesis to further assist them control their credit default risk profile.

Risks that build-up unchecked (from one or more of poor identification, quantification or control) within individual institutions or wider financial system may lead to insolvency of an institution and/or financial system instability. Credit risk is the largest risk credit-granting institutions face (Apostolik et al., 2009), thereby attracting significant attention from both: (1) management who wish to operate profitably on behalf of their stakeholders by

maintaining strong capitalisation and adequate provisions for credit losses; and (2) regulators who aim to achieve a stable, competitive and functioning banking system so institutions fulfil financial promises they make. It is therefore important credit risk is measured properly and appropriately by each credit-granting institution. In Australian, the Australian Prudential Regulation Authority (APRA) supervises all authorised deposit-taking institutions (banks, building societies and credit unions) with a mission to "…ensure that in all reasonable circumstances, financial promises made by institutions we supervise are met within a stable, efficient and competitive financial system" (APRA, 2020).

For further perspective and motivation why credit risk management is important, consider that total loans and advances in Australia as at July 2020 stood at \$2.8 trillion across all 127 licensed Authorised Deposit-Taking Institutions (APRA, 2020). This exceeds Australia's Gross Domestic Product of \$1.9 trillion for year ending December 2019 (ABS, 2020), which represents the total value added over a twelve-month period across all economic activity within the entire Australian economy. On top of this, concentration risks exist in the Australian banking sector, as these 127 Authorised Deposit-Taking Institutions comprise few large and many small firms. The four largest comprise over 75% of all loans and advances, the seven largest over 81%, and the 20 largest over 96% (APRA, 2020). Further, home loans dominate lending for these largest seven lenders, as per Figure 1.1. The Global Financial Crisis of 2008 demonstrated the knock-on effect of how the failure of individual institutions (via either nationalisation, bankruptcy, or consolidation) led to the ensuing spread of systematic instability in the wider financial system and ultimately protracted recessions in many economies. Thus clear measurement and understanding of the risk of credit default, particularly for home loans, are important for an individual institutions, financial system stability, economies and ultimately society.



Figure 1.1: Loans and Advances for Australia's Seven Largest Lenders (as at August 2020, Source: APRA (2020))

1.1.2 Expected Loss (EL) and Unexpected Loss (UL)

Institutions manage their loan portfolio anticipating some non-zero amount of credit losses, called "expected loss" (EL), which is viewed as a cost of doing business. Institutions estimate expected loss and accordingly set prices charged to customers and hold on their balance sheet an asset called a loan-loss provision in anticipation the expected losses materialise. The loss experience institutions encounter inevitably varies year-to-year from that anticipated, with losses greater than the expected loss called the "unexpected loss" (UL). Institutions estimate this unexpected loss and accordingly hold capital as a buffer to absorb these should they occur. The link between the variability of losses over time, and how the variation beyond the expected loss results in unexpected loss are demonstrated in figure 1.2.



Figure 1.2: Relationship Between Expected Loss and Unexpected Loss. Source:BIS (2005)

As a scarce resource, infinite capital cannot be held to absorb losses the institution may make in every eventuality. Therefore a trade-off is struck by selecting a very high confidence level (prescribed as 99.9% under the Basel Accords (BIS, 2006)) from the credit loss distribution for which capital will be held. This is the credit "Value at Risk" or "VaR" (see for example Jorion (2006)), representing the largest loss with 99.9% confidence that can be weathered by the institutions over a 1-year period. Losses beyond this VaR point will lead to the institution's insolvency. The relationship between "expected loss", "unexpected loss" and the VaR point are displayed in figure 1.3.



Potential credit losses

Figure 1.3: Credit Loss Distribution. Source: Yeh et al. (2005)

1.1.3 Credit Risk Components - PD, EAD, LGD

Components of "expected loss" are

- "Probability of Default" (PD) The probability a customer will fail to make full and timely contractual repayments and default (this is the risk of credit default);
- "Exposure At Default" (EAD) The expected value of the loan at the time of default;
- "Loss Given Default" (LGD) The amount the institution will likely lose if the customer defaults, as a percentage of EAD.

This thesis focuses on quantitative estimation of PD. See for example Thackham and Ma (2019) and Tong et al. (2013) for example background on EAD and LGD respectively.

"Expected loss" is calculated as the product of frequency of credit default (PD) and the severity of loss for defaulted loans (EAD times LGD) (Apostolik et al., 2009):

$$EL = \underbrace{PD}_{\text{frequency}} \times \underbrace{EAD \times LGD}_{\text{severity}}.$$
 (1.1)

"Unexpected loss" is calculated using the Vasicek (2002) Asymptotic Single Risk Factor Model prescribed in the Basel Accords (BIS, 2006)

$$UL = \left[\Phi\left(\sqrt{\frac{1}{1-R}}\Phi^{-1}(PD) + \sqrt{\frac{R}{1-R}}\Phi^{-1}(0.999)\right) - PD\right] \times dEAD \times dLGD \quad (1.2)$$

where *R* represents the correlation between each loan and a common latent factor, and dLGD and dEAD are the LGD associated with an economic downturn. See for example Rutkowski and Tarca (2015) which discuss the Brownian motion construct using a single-factor copula underlying equation (1.2).

1.1.4 Probability of Default (PD)

Institutions approved by their regulator to use the Advanced Internal Ratings Based (A-IRB) Approach for their minimum credit capital requirement under the Basel Accords (BIS, 2006) can employ advanced modelling techniques trained on their own data to estimate PD, EAD and dLGD necessary as inputs to equation (1.2). Institutions also employ advanced

modelling techniques to estimate PD, EAD and LGD as inputs to equation (1.1) required under international accounting standards (IASB (2014), FASB (2016)). Models to estimate customer default find further additional uses in wider risk management applications, covering: loan underwriting (which customers to lend to); loan pricing (how much to charge customers); risk appetite (how many loans to write and which types of customers to write them to); and profit analysis (how much profit was actually earned from loans that were written) (Siddiqi, 2005). Institutions that poorly manage risks may face heightened regulatory scrutiny, increased cost of funding, or even being wound-up – each of these put in jeopardy the ability of an institution to keep its financial promises. Therefore accurate estimates that dynamically explain the risk of credit default are critical to an institution's quantitative credit risk management.

1.1.5 Estimating Credit Default using Survival Analysis

For models estimating the risk of credit default to meet internal and external approval, model drivers and functional form need to be intuitive and explainable. Consequently regression models are often employed with the preferred approach being logistic regression (Baesens et al. (2003), Lessmann et al. (2015). Multiple studies (see for example Bellotti and Crook (2009), Stepanova and Thomas (2002)) outline that while both logistic regression and survival analysis achieve comparable accuracy, Tong et al. (2012) details key advantages of survival analysis, being: (1) inclusion of censored data (logistic regression removes partially observed data); and (2) ready production of default probabilities over any desired time period (logistic regression predicts only over a single fixed timer period).

Survival analysis are statistical techniques to model the length of time until an event occurs. Subjects are followed in anticipation they encounter an event, with these techniques applicable in fields like medicine, industrial life testing and finance (Kalbfleisch and Prentice (2002), Lessmann et al. (2015)). Survival analysis provides an intuitive modelling approach using available covariates to help explain variation in time to event, also known as "survival time" (Hosmer et al., 2008). Chief amongst these techniques is the semi-parametric Cox model (1972, 1975), which introduced the seminal approach of partial likelihood for regression coefficient estimation and remains a popular technique (Ren and Zhou, 2011).

Model extensions include catering for time varying covariates (Crowley and Hu, 1977) and mixture-cure semi-parametric Cox models (Sy and Taylor, 2000).

Narain (1992) first introduced survival analysis to estimating the risk of credit default, with many more authors since (see for example Bellotti and Crook (2009)) outlining how the semi-parametric Cox model is an applicable technique to estimate the the risk of credit default. Credit-granting institutions sell loans to customers, with these customers entering a contractual obligation to repay all monies owed, specifically principal, interest and fees. Examples are credit cards and home loan mortgages granted to individuals, and corporate loans granted to businesses. Some customers fail to repay in accordance with their agreed credit contract and thereby default on their obligations. In this setting, institutions follow their customers in anticipation of a default event occurring, allowing the "survival" to be statistically modelled, where the meaning of "survival" involves the time until a credit default. This is exemplified in Figure 1.4 below in a stylisation of the survival function for the risk of credit default.



Figure 1.4: Stylised Example of the Application of Survival Analysis to Credit Default

1.2 Aims and Contributions

1.2.1 Existing Approaches and Limitations

Estimating the risk of credit default is critically intertwined with the competing risk a customer fully repays their loan while also having key predictive drivers with values that change over time (Wycinka, 2019). Logistic regression cannot incorporate non-binary outcomes but could be substituted for multinomial regression. Further, while logistic regression can incorporate panel data for drivers that change over time, Hayden and Porath (2011) point out credit default modelling practitioners typically ignore the dynamic pattern of the covariates and simply fit the model assuming cross-sectional data. Erlenmaier (2011) describes this as the "cohort method" which observes the number of customers at the beginning of each period (usually a year) and the number of customers that default during the ensuing year. Tong et al. (2012) outlines how, for example, a three-year loan to a single customer would then give rise to three separate records in the data but the drivers whose values change over time potentially are different for each three records. A consequence of ignoring the panel data structure when fitting a cross-sectional logistic regressions are that variance estimates may be incorrect.

For these reasons, survival analysis is gaining popularity in credit risk modelling, with competing risk semi-parametric Cox Model and mixture-cure semi-parametric Cox Model being conventional approaches. However partial likelihood involved in parameter estimation of these models has two shortcomings: (1) the baseline hazard is not estimated, so calculating probabilities requires a further estimation step; and (2) a covariance matrix for both regression coefficients and the baseline hazard is not readily produced. This thesis adds to the available literature by addressing these computational aspects of the semi-parametric Cox model and testing these via both extensive simulations as well as estimating models to predict credit default for home loans. These enhancements are important in applied credit risk modelling as they provide a clear functional form of how risk drivers impact the risk of credit default and produce asymptotic correct variance estimates for all model parameters.

1.2.2 Summary of Contributions - 2 Publications, 1 Manuscript

This thesis is arranged by starting with a literature review in Chapter 2 that introduces the semi-parametric Cox model outlining how the model can be applied to explain the risk of credit default. The three subsequent chapters embody the research of this thesis. Chapter 3 is a published paper (Thackham and Ma, 2020a) that extends the work of Ma et al. (2014) to estimate the semi-parametric Cox model with time-varying covariates using maximum likelihood. Chapter 4 is a published paper (Thackham and Ma, 2020b) that extends Thackham and Ma (2020a) and Ma et al. (2014) to estimate competing risk semi-parametric Cox model with time-varying covariates. Works in chapters 3 and 4 both employ maximum likelihood with constrained optimisation to jointly estimate regression coefficients and baseline hazard, where the constraint ensures the baseline hazard non-negativity. Not only does including time varying covariates open up a richer vein of potential covariates, but their inclusion is also important for modelling the risk of credit default as estimates need to be flexible enough to take into account changes in the risk profile of customer behaviour over time. Examples of such variables usually considered are: (1) episodes of recent customer delinquency; and (2) up-to-date valuations of collateral and loan balances as measured via the loan-to-value ratio. These variables by design change over time so therefore cannot be captured simply as baseline covariates. Both chapters 3 and 4 prove the asymptotic properties of the estimators. Additionally, work in chapter 4 specifically incorporates competing risk of early loan closure and the work goes on to estimate the expected number of defaults at one-year intervals over the ten years of observed data. Chapter 5 is a manuscript submitted in consideration for publication that extends the work of Ma et al. (2014) to estimate the mixture-cure semiparametric Cox model using maximum likelihood with constrained optimisation. The work in chapter 5 jointly estimates parameters by maximising a likelihood function, with asymptotic variances for all model parameters readily returned and available for drawing inferences.

All the methods in chapters 3, 4 and 5 are implemented in R (with code included in the appendix of this thesis), allowing them to be demonstrated via a comprehensive simulation exercise. When applied to a real-world credit risk dataset, the methods clearly produce intuitive parameter estimates of key drivers of credit default, covering baseline covariates (such as product type and customer age) as well as key time-varying covariates (such as dynamic loan-to-value ratio and recent delinquency behaviour). Additionally, the methods

estimate the baseline hazards and associated asymptotic variance. This allows confirmation in the given data of the "humped-hazard", a previously reported feature (see for example Im et al. (2012) and Bellotti and Crook (2013)), whereby the risk of credit default begins lower, rises to a peak, before falling again afterwards.

2

Literature Review

This literature review chapter contains four sections. The first introduces survival data and survival analysis in the context for modelling the risk of credit default. The second is a more detailed literature review of the models explored in-depth in chapters 3, 4 and 5, focussing on the semi-parametric Cox model (1972, 1975). The third is a literature review specifically focussed on application of survival analysis to credit default modelling. Starting with Narain (1992) who introduced survival analysis to credit risk modelling, the section canvasses subsequent contributions by numerous authors, with a focus on those adopting time-varying covariates, competing risk approaches and mixture-cure approaches. The fourth describes the structure of the remaining chapters of this thesis.

2.1 Background - Survival Data and Survival Analysis

This section details how credit default data can be analysed using the semi-parametric Cox models explored in this thesis. It details survival data must unambiguously define three elements: (1) an origin; (2) a time scale, and (3) an event(s) of interest. Uninformative right-censoring is introduced. It also introduces applications of semi-parametric Cox model investigated in this thesis: (1) single-event survival analysis; (2) competing risk survival analysis; and (3) mixture-cure survival analysis.

2.1.1 Survival Data

Survival analysis observes *subjects* spanning *time* until an *event* occurs. Cox and Oakes (1984) state survival data requires three elements:

- 1. Time origin, when the subject became at risk of an event;
- 2. Time scale, measuring the passage of time (days, weeks, months, years); and
- 3. Event, or set of events, clearly defined.

A credit-granting institution that writes a loan faces the risk the customer does not make contractual repayments, thereby defaulting. In this case:

- 1. the event is credit default;
- 2. the time origin begins when the loan contract starts; and
- 3. the **time scale** is months since start of the loan contract.

Censoring, a type of missingness common in survival data (Klein and Moeschberger, 2003), arises when an event for a subject is only known to have occurred during a specified period rather than being precisely known. Censoring needs careful consideration otherwise the results may be biased (Sterne et al., 2009). Three types of censoring which can vary between subjects are:

- Right subject remains at risk of an event at a certain point. This is further split:
 - Type I random drop-out and/or administrative closure of the study.
 - Type II study closes when a fixed number of subjects encounter an event.
- Left subject experienced an event prior to entrance to the study.
- Interval subject experienced event during a known interval, but the precise time is unknown.

Klein and Moeschberger (2003) state censoring is often random (referred to as "noninformative") type I right censoring. This assumes right-censoring times are independent of survival times, so that for each subject there is a right censoring time and the survival time is observed if and only if the survival time is less than the censoring time. If the survival time is greater than the censoring time, then the subject has left the study without having an event of interest observed and the time for this subject is recorded as as the censoring time. While there are approaches to cater for complex censoring, given that right-censoring is in practice the type most commonly encountered modelling the risk of credit default, this thesis considers only right-censoring. Future research could include extension to (for example) informative right censoring and interval censoring.

Truncation is a deliberate sample design choice which determines which subjects enter the study. In credit risk, left truncation (also called "delayed entry") arises when customers enter the study after a period of delay from the origin (Kalbfleisch and Prentice, 2002). This is not in scope for this thesis, but does represent an avenue of future research.

2.1.2 Survival Analysis

This thesis explores three applications of the semi-parametric Cox model to analyse credit default: (1) single-event analysis; (2) competing risk analysis; and (3) mixture-cure analysis. All three study the same random variable (time to event) and share similar input data structure (requiring an origin, scale and event(s)). These three applications of the semi-parametric Cox model studied in this thesis are:

- 1. Semi-parametric Cox Model assumes all customers default if followed for sufficient length of time, and customers not observed to default are assumed to be uninformatively right-censored. See Figure (2.1).
- 2. Semi-parametric Cause-Specific Cox Model with Competing Risks assumes all customers will undergo at most one of either default or repayment (known the competing risks) and when they do so all the remaining risks are impossible to encounter there after. Further customers not observed to either default or repay are assumed to be uninformatively right-censored. See Figure (2.2).
- 3. Semi-parametric mixture-cure Cox Model assumes some fraction of customers are

susceptible to credit default, and customers not observed to default are assumed to be either from the non-susceptible population or uninformatively right-censored. See Figure (2.3).



Figure 2.1: Semi-Parametric Cox Model Applied to Credit Default Risk



Figure 2.2: Semi-Parametric Cause-Specific Cox Model with Competing Risks Applied to Credit Default Risk



Figure 2.3: Semi-Parametric Mixture-Cure Cox Model Applied to Credit Default Risk

2.2 Literature Review - Models

This section is a detailed literature review of: (1) survival analysis; (2) competing risk analysis; and (3) mixture cure analysis. Each is covered in individual subsections, tracing the origin of the analysis (in some cases the 17th century) before detailing the semi-parametric Cox Model this thesis applies to credit risk modelling.

2.2.1 Single-Event Survival Analysis

Single-event survival analysis models the time to a single event of interest. Censored timeto-event data was first studied by actuaries (Fisher and Lin, 1999), with English astronomer Edmund Halley (1693) developing the first life table to describe human mortality. Several related functions describe the distribution of a random variable *T* representing time to event (see for example Klein and Moeschberger (2003)), where knowledge of one function allows recovery of the others. These are the: survival function S(t); the hazard function h(t); and cumulative hazard function H(t). The survival function (the complement of the cumulative distribution function F(t)) represents the probability a subject survives beyond time *t*

$$S(t) = P[T > t] = 1 - F(t) = 1 - \int_0^t f(u) du$$
(2.1)

where f(t) is the density function of *T*. The hazard function h(t) is fundamental to survival analysis playing a central role in the semi-parametric Cox model (1972, 1975). Representing the instantaneous probability of an event occurring, conditional on the subject not experiencing the event yet, mathematically it is

$$h(t) = \lim_{\Delta t \to 0} \frac{P[t < T \le t + \Delta t | T > t]}{\Delta t} = \frac{f(t)}{S(t)}.$$
(2.2)

The cumulative hazard function is $H(t) = \int_0^t h(u) du$.

These functions can be extended to compare survival between two or more groups or conditionally explaining survival in a regression using covariates. There are broadly three categories of analysis techniques (see for example Kalbfleisch and Prentice (2002) and Hosmer et al. (2008)) which are outlined below. Dirick et al. (2017) details in a benchmark study how these techniques are applied to credit default modelling.

The first are non-parametric techniques for which no functional form is assumed for the distribution *T*. Prominent methods include the Kaplan-Meier (1958) survival function estimator which estimates S(t) and the Nelson-Aalen cumulative hazard estimator (Nelson (1969, 1972) and Aalen (1978)) which estimates H(t). Tests for significant differences in survival between groups can be undertaken using (for example) the log-rank test (Mantel (1966), Peto and Peto (1972)). Non-parametric techniques are not covered by this thesis. The second are parametric techniques (which allow inclusion of covariates) and further assume a functional form for both covariates and the distribution of T using a regression. Called accelerated failure time (AFT) models, these assume (for example) either a Weibull, exponential, gamma, log-logistic or log-normal distribution for T (Kalbfleisch and Prentice, 2002). Similarly, parametric techniques are not covered by this thesis.

The third are semi-parametric models (which allow for the inclusion of covariates), and assume a functional form for the covariates via a regression, but make no additional assumptions for the distribution of T. The semi-parametric Cox model (1972, 1975) is a widely used semi-parametric approach to estimate the hazard h(t) and is the primary focus of this thesis, not only for single-event survival analysis but also for competing risk analysis and mixture-cure analysis. Hosmer et al. (2008) cites the attraction of such regression techniques is that plausible models may be easily fit, evaluated, and interpreted. Kalbfleisch and Prentice (2002) describe one of its chief benefits is that it allows an intuitive explanation of the hazard conditional on explanatory covariates.

2.2.1.1 Semi-Parametric Cox Model - Baseline Covariates

The semi-parametric Cox (1972, 1975) model is the corner-stone of modern survival analysis (Zheng and Lin, 2007). Let T_i be the event time for subject *i* and C_i the corresponding non-informative right censoring time so that the observable survival time is $Y_i = \min(T_i, C_i)$. Denoting the observed Y_i by y_i (i = 1, ..., n), each y_i is then either a time recording an event of interest ($\delta_i = 1$) or censoring time ($\delta_i = 0$). Additionally, suppose there are *p* explanatory covariates arranged in the vector $\mathbf{x}_i^T = [x_1, ..., x_p]$ available for each subject (i = 1, ..., n) which are thought to explain time to event. These explanatory variables can include quantitative variables (such as customer age or loan balance), qualitative variables (such as treatment group or product type) as well as potentially interactions between covariates. The observed data becomes ($y_i, \mathbf{x}_i, \delta_i$), and can be used to estimate the semi-parametric Cox model

$$h_i(t|\mathbf{x}_i) = h_0(t)e^{\mathbf{x}_i^T\boldsymbol{\beta}} \tag{2.3}$$

where $\boldsymbol{\beta}^{T} = [\beta_{1}, ..., \beta_{p}]$ are regression coefficients for the covariate vector \boldsymbol{x}_{i}^{T} , and $h_{0}(t)$ is an arbitrary unspecified function of time called the baseline hazard that combines multiplicatively with the covariate effects to act on the hazard function. This leads to the conditional survival function $S_{i}(t|\boldsymbol{x}_{i}) = S_{0}(t)^{[e^{x_{i}^{T}\beta}]}$, where $S_{0}(t)$ is the baseline survival function.

With only baseline (fixed-time) covariates, the Cox model is commonly referred to as the "proportional hazards" model, because hazards between subjects are proportional (Kalbfleisch and Prentice, 2002). That is, the ratio of hazards between two different subjects *A* and *B*

$$\frac{h_A(t|\mathbf{x}_A)}{h_B(t|\mathbf{x}_B)} = \frac{h_0(t)e^{\mathbf{x}_A^T \boldsymbol{\beta}}}{h_0(t)e^{\mathbf{x}_B^T \boldsymbol{\beta}}} = e^{(\mathbf{x}_A^T - \mathbf{x}_B^T)\boldsymbol{\beta}}$$
(2.4)

is constant over time. This value is called the relative risk (Klein and Moeschberger, 2003) or the hazard ratio (Hosmer et al., 2008). It is important to test that the assumption of proportional hazards holds when applying the Cox model to a dataset. This can be conducted by (for example) scaled Schoenfeld (1981) residuals or by plotting log-log survival curves (Cox and Oakes, 1984). If these tests detect a violation of the proportional hazards assumption, then one remedy can be to include a time-varying covariate in the model (Hosmer et al., 2008).

To estimate the effects $\boldsymbol{\beta}^T = [\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_p]$ in equation (2.3), using observed data $(y_i, \boldsymbol{x}_i, \delta_i)$ (i = 1, ..., n), one can use Cox's partial likelihood (Cox 1972, 1975). The Cox partial likelihood can be derived as a profile likelihood using the likelihood adapted for right-censored data (Klein and Moeschberger (2003), Johansen (1983)).

Rather than detailing the derivation of the partial likelihood, we instead outline an intuitive derivation from Cox and Oakes (1984). In the absence of tied failure times, let $\tau_1 < \tau_2 < ... < \tau_d$ be the ordered failure times amongst the *n* subjects in the sample, where *d* is the total number of subjects in the sample ever observed to experience the event of interest and (n - d) is the total number of censored subjects. Let $\Re(\tau_j)$ be the risk set which is comprises all subjects still under observation in the study (ie: either not yet censored or not yet encountered an event of interest) just before the *j*th failure time τ_j , (j = 1, ..., d). As per Klein and Moeschberger (2003), the conditional probability that subject *i* fails at τ_j with covariates \mathbf{x}_j^T

given that one individual fails from the risk set $\mathscr{R}(\tau_j)$ is

$$P[\text{subject } i \text{ fails at } \tau_j \mid \text{one failure at } \tau_j]$$

$$= \frac{P[\text{subject } i \text{ fails at } \tau_j \mid \text{subject } i \text{ survives to } \tau_j]}{P[\text{one failure at } \tau_j \mid \text{subject } i \text{ survives to } \tau_j]}$$
(2.5)
$$= \frac{e^{x_i^T \beta}}{\sum_{k \in \mathscr{R}(\tau_j)} e^{x_k^T \beta}}.$$

The partial likelihood is formed by multiplying these conditional probabilities over all d observed failure times from the sample. For (i = 1, ..., n), where n is the sample size of all subjects, the resulting partial likelihood function is

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \left[\frac{e^{\boldsymbol{x}_{i}^{T}\boldsymbol{\beta}}}{\sum_{k \in \mathscr{R}(\tau_{j})} e^{\boldsymbol{x}_{k}^{T}\boldsymbol{\beta}}} \right]^{\delta_{i}} = \prod_{i=1}^{d} \left[\frac{e^{\boldsymbol{x}_{j}^{T}\boldsymbol{\beta}}}{\sum_{k \in \mathscr{R}(\tau_{j})} e^{\boldsymbol{x}_{k}^{T}\boldsymbol{\beta}}} \right]$$
(2.6)

where $\delta_i = 1$ for subjects with an observed failure time, such that $\sum_{i=1}^{n} \delta_i = d$. The numerator depends only on information from the individual subject who experiences an event of interest at τ_j , whereas the denominator captures information from all subjects in the risk set (ie: those who have not yet experienced an event and/or have not yet been censored). Estimation is carried out using the log-partial likelihood function

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \delta_i \left[\boldsymbol{x}_i^T \boldsymbol{\beta} - \ln \left(\sum_{k \in \mathscr{R}(\tau_i)} e^{\boldsymbol{x}_k^T \boldsymbol{\beta}} \right) \right].$$
(2.7)

The maximum partial likelihood estimator, denoted as $\hat{\beta}$, is obtained by differentiating equation (2.7) with respect to β and solving $\partial l(\beta)/\partial \beta = 0$ (Hosmer et al., 2008). The variance estimator is obtained in a similar manner for maximum likelihood, as the inverse of the negative second order derivative of the log-partial likelihood (i.e.: the inverse of the observed information $I(\beta)$ from the partial likelihood). Klein and Moeschberger (2003) discuss how these equations are amended for tied survival times.

To estimate the baseline hazard $h_0(t)$ there are two common estimators available, each additional estimation step requiring estimated regression coefficients $\hat{\beta}$ from the semiparametric Cox model as input. The first is the Breslow (1972) estimator of the cumulative baseline hazard function $H_0(t)$, which can be derived by maximising a profile likelihood conditional on the log-hazard ratio estimates. This estimator has the undesirable feature that it produces point estimates of $h_0(t)$ that are very "noisy" and unstable (Hosmer et al., 2008). This estimator becomes the non-parametric Nelson-Aalen estimator of the cumulative hazard when there are no covariates present (Klein and Moeschberger, 2003). The second is the Kalbfleisch and Prentice (2002) estimator of the baseline survival function $S_0(t)$ which mirrors the derivation of the Kapaln-Meier non-parametric estimator and Rodriguez (2005) outlines how the estimator is constructed. The estimator also produces "noisy" and unstable point estimates of $h_0(t)$. It reduces to the non-parametric Kaplan-Meier estimator when no covariates are present (Hosmer et al., 2008).

2.2.1.2 Semi-Parametric Cox Model - Time Varying Covariates

Often there are covariates whose value for a given subject may change over time during their observation in the study (Cox and Oakes, 1984). Crowley and Hu (1977) extend the semi-parametric Cox model (1975) to include time-varying (also known as time-dependent) covariates. These differ fundamentally from baseline covariates, whose values are measured only once for each subject at entry to the study and either do not change or do change but are not tracked over time. Time-varying covariates are very important for modelling credit default (Bellotti and Crook, 2014), with variables such as dynamic loan-to-value ratio, recent customer delinquency and loan top-ups all important for predicting home loan default (see for example Thackham and Ma (2020a) and Thackham and Ma (2020b)).

Crowley and Hu (1977) extend the Cox model to cater for time-varying covariates, analysing the famous Stanford heart transplant data. Their model uses (in addition to baseline covariates) transplant status and transplant age to predict survival of patients. The method developed estimates the regression coefficients for both baseline and time-varying covariates using an amended version of the partial likelihood. The amendment allows the same subject to have potentially different values for their covariates in different risk sets. In addition to baseline covariates $\mathbf{x}_i^T = [x_1, ..., x_p]$, let $z_i(t)^T = [z_1(t), ..., z_q(t)]$ be a tuple of q time varying covariates for the i^{th} subject. This assumes that the covariate process $z_i^T(t)$ is known for any time which the subject is under observation (Kalbfleisch and Prentice, 2002). However Therneau et al. (2015) point out that the partial likelihood only requires values of $z_i^T(t)$ to be precisely known at the d failure times. Following equation (9.2.1) of Klein and Moeschberger

(2003), for (i = 1, ..., n), where *n* is the sample size of all subjects, the available data becomes $(y_i, x_i, z_i(t), \delta_i)$. If event times are distinct and $\tau_1 < \tau_2 < ... < \tau_d$ denotes these ordered event times, $x_{(j)}$ and $z_{(j)}(\tau_j)$ denotes the covariates associated with subject whose event time τ_j and $R(\mathscr{R}(\tau_j))$ is the risk set at time τ_j , then the partial likelihood including time-varying covariates $z_i^T(t)$ with regression coefficients γ is

$$L(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \prod_{j=1}^{d} \left[\frac{e^{\boldsymbol{x}_{(j)}^{T} \boldsymbol{\beta} + \boldsymbol{z}_{(j)}^{T}(\tau_{j}) \boldsymbol{\gamma}}}{\sum_{k \in \mathscr{R}(\tau_{j})} e^{\boldsymbol{x}_{k}^{T} \boldsymbol{\beta} + \boldsymbol{z}_{k}^{T}(\tau_{k}) \boldsymbol{\gamma}}} \right]$$
(2.8)

where $\delta_i = 1$ for subjects with an observed failure time, such that $\sum_{i=1}^{n} \delta_i = d$. Parameter estimation and inference proceeds similarly as for the case with only baseline covariates.

2.2.1.3 Baseline Hazard Estimation

In many applications of the Cox model, such as modelling credit default, estimation of survival probabilities (rather than simply just the regression coefficients) are of interest. Royston (2011) opines this should entail an explicit estimate of the baseline hazard function $h_0(t)$. For example, van Houwelingen (2000) states:

"It is the duty of the [statistician] involved in reporting the prognostic model to give all the information needed to build further on their model. For Cox models that should also include the baseline hazard or survival rate, if possible smoothed somehow or given in an approximate functional form using fractional polynomials, exponentials, rational functions or something similar."

Given that the partial likelihood specifically does not estimate the baseline hazard $h_0(t)$, recovery of survival probabilities requires an additional estimation step. Incorporating both baseline covariates \mathbf{x}_i^T and time-varying covariates $\mathbf{z}_i^T(t)$, the survival function is $S(t) = e^{-H(t)}$, where

$$H(t) = \int_0^t h_0(u)^{exp\left(\mathbf{x}_i^T\beta + \mathbf{z}_i^T(u)\boldsymbol{\gamma}\right)} du.$$
(2.9)

There are more recent developments of the baseline hazard estimators that in general rely on a log-transform of $h_0(t)$. Royston (2011) proposes a method to approximate the

log-baseline hazard (which ensures positivity of the baseline hazard) using fractional polynomials and restricted cubic splines. However, similar to the Breslow (1972) and Kalbfleisch and Prentice (2002) methods, Royston's method requires as input the estimated regression coefficients from the partial likelihood method. This means that Royston (2011) does not attempt to estimate both the effects and the baseline hazard jointly.

Cai and Betensky (2003) estimating the log-hazard using linear splines for interval censored data. Cai et al. (2012) extend this to estimate the baseline hazard using linear splines to model the log-hazard (which ensures positivity of the hazard) with smoothing parameters estimated by restricted maximum likelihood (REML). The authors recast the problem as a mixed-effects Poisson regression with an offset term, which allows estimation in standard statistical packages, such as SAS or R. Their methodology does not cater for conditional explanation of survival times using covariates.

Kneib and Fahrmeir (2004) provide several extensions to the Cox model, calling their model a mixed-hazard regression. Their extensions include modelling log-baseline hazard (to ensure positivity of the baseline hazard) using penalised splines, as well as allowing for time-varying covariates.

2.2.1.4 Maximum Penalised Likelihood Estimation

Ma et al. (2014) develop an approach to simultaneously estimate the regression coefficients and the baseline hazard (without a log-transform) for the Cox model that contains non-timevarying covariates, an approach this thesis extends. The approach of Ma et al. (2014) uses maximum penalised likelihood (MPL), where a penalty is used to impose a degree of smoothness to the baseline hazard. The parameters are estimated using constrained optimisation to respect the non-negativity of the baseline hazard. Starting with the hazard $h_i(t) = h_0(t)e^{x_i^T\beta}$, the likelihood for all subjects (i = 1, ..., n) is

$$L(\boldsymbol{\beta}, h_0(t)) = \prod_{i=1}^n L_i(\boldsymbol{\beta}, h_0(t))$$
(2.10)
where for the *i*th subject with observed time y_i and censoring indicator δ_i , we have $L_i(\boldsymbol{\beta}, h_0(t_i)) = [f_i(t_i)]^{\delta_i} \times [S_i(t_i)]^{(1-\delta_i)}$. The log-likelihood is then

$$l(\boldsymbol{\beta}, h_0(t)) = \log \left(L(\boldsymbol{\beta}, h_0) \right) = \sum_{i=1}^n \delta_i \log \left(f_i(y_i) \right) + (1 - \delta_i) \ln \left(S_i(y_i) \right).$$
(2.11)

Substituting into equation (2.11) the fact that $\log (S_i(y_i)) = -H_i(y_i)$ and $f_i(y_i) = h_i(y_i)S_i(y_i)$, the log-likelihood becomes:

$$l(\boldsymbol{\beta}, h_0(t)) = \sum_{i=1}^n \delta_i \left[\log (h_i(y_i)) + \log (S_i(y_i)) \right] + (1 - \delta_i) \log (S_i(y_i))$$

= $-\sum_{i=1}^n H_i(y_i) + \sum_{i=1}^n \delta_i \log (h_i(y_i))$ (2.12)

The method caters for baseline covariates x_i^T , which introduced to equation (2.12) the loglikelihood becomes

$$l(\boldsymbol{\beta}, h_0(t)) = -\sum_{i=1}^n H_0(y_i) e^{\mathbf{x}_i^T \boldsymbol{\beta}} + \sum_{i=1}^n \delta_i (\log (h_0(y_i)) + \mathbf{x}_i^T \boldsymbol{\beta}).$$
(2.13)

Because $h_0(t)$ has infinite dimension, the authors introduce a basis function to approximate the baseline hazard $h_0(t) = \sum_{u=1}^{m} \theta_u \psi_u(t)$, where $\psi_u(t)$ is the basis function and θ_u are values that require estimation when fitting the model. The authors impose two conditions on model fitting: (1) a smoothness constraint by adding a penalty term to the log-likelihood, denoted $\lambda J(h_0(t))$; and (2) requiring all values of θ_u to be greater than or equal to zero. This results in needing to undertake constrained maximisation of the penalised log-likelihood function

$$\Phi(\beta, h_0(t)) = l(\beta, h_0(t)) - \lambda J(h_0(t)).$$
(2.14)

Defining the basis for the cumulative baseline hazard as $\Psi_u(t_i) = \int_0^t \psi_u(v) dv$, the cumulative hazard is written as:

$$H_0(t) = \int_0^t h_0(s) ds = \sum_{u=1}^m \theta_u \int_0^t \psi_0(s) ds = \sum_{u=1}^m \theta_u \Psi_u(t).$$
(2.15)

This results in the penalised log-likelihood

$$\Phi(\boldsymbol{\beta}, h_0(t)) = -\sum_{i=1}^n \sum_{u=1}^m \theta_u e^{\boldsymbol{x}_i^T \boldsymbol{\beta}} \Psi_u(t) + \sum_{i=1}^n \delta_i \left(\log \left(\sum_{u=1}^m \theta_u \psi_u(t) \right) + \boldsymbol{x}_i^T \boldsymbol{\beta} \right) - \lambda J(h_0(t)). \quad (2.16)$$

In the two published papers (chapter 3 and chapter 4) and a prepared manuscript (chapter 5) of this thesis extends the work of Ma et al. (2014) and investigates results using both simulation and application to modelling the risk of credit default. Chapter 3 extends the work to include both baseline and time-varying covariates. Chapter 4 further extends the work to include competing risks (for default and repayment) using both baseline and time-varying covariates. Chapter 5 extends the work to mixture-cure model with baseline covariates.

2.2.2 Competing Risk Analysis

Competing risk models analyse the time until subjects encounter one of potentially many possible risks. "Competing" means risks for each subject are mutually exclusive so the occurrence of an event precludes any other event from occurring for that subject. For example if the event of primary interest is home loan default then a competing risk is early repayment. Competing risks analysis, also called "multiple-decrements analysis" by actuaries (Deshmukh, 2012), has a similar genesis as survival analysis (David and Moeschberger (1978)). The first recorded use was by Daniel Bernoulli (1760) who extended Edmund Halley's (1693) method for constructing life tables to demonstrate the increase in human survival if the competing risk of small pox were eliminated as a cause of death.

Often competing risks are ignored and studies focus on the primary risk by simply treating competing events as censored (Austin et al., 2016). This can lead to problems, for example Putter et al. (2007) demonstrates when applying the Kaplan-Meier (1958) estimate to the event of interest while treating all other competing risks as censored underestimates the survival function of the event of primary interest.

To start, let T_i be the true survival time (which may not be observed due to censoring), and C_i the corresponding non-informative right-censoring time, thus the observed survival time

is $Y_i = \min(T_i, C_i)$. The value of T_i for each subject is associated with g = 1, ..., G mutually excluding competing risks. Denoting realised values of Y_i with y_i , the tuple of observed values for each subject becomes $(y_i, \delta_{i1}, ..., \delta_{iG})$ where $\delta_{ig} = 1$ when the subject encounters risk g and zero otherwise. Additionally $\delta_{i0} = \sum_{g=1}^{G} \delta_{ig}$ so $\delta_{i0} = 0$ when the event time for a subject is censored.

An important function for competing risk analysis is the Cumulative Incidence Function (CIF). Pintilie (2006) outlines that the CIF for event type g represents the probability that event g occurs before time t in the presence of all other possible causes, and is defined as

$$F_g(t) = P[Y_i \le t, g].$$
 (2.17)

The Cumulative Distribution Function (CDF) for any of the g events to occur before time t is the sum of the g individual CIFs

$$F(t) = P[Y_i \le t] = \sum_{g=1}^G F_g(t) = \sum_{g=1}^G P[Y_i \le t, g].$$
(2.18)

For the g^{th} competing risk, Pintilie (2006) details the cause-specific hazard is

$$h_g(t) = \lim_{\Delta t \to 0} \frac{P[t < Y_i \le t + \Delta t, G = g|Y_i > t]}{\Delta t} = \frac{f_g(t)}{S(t)}$$
(2.19)

where S(t) is the all-risk survival, that is the survival of a subject not encountering any of the (g = 1, ..., G) risks prior to time t. The cause specific hazard does not have direct interpretation in terms of survival probabilities for the particular event type. In addition, the following relationships hold for h(t), H(t), f(t), F(t) and S(t) which respectively are functions for the hazard, cumulative hazard, density, cumulative distribution and survival for all risk combined

$$h(t) = \sum_{g=1}^{G} h_g(t)$$
 (2.20)

$$H(t) = \sum_{g=1}^{G} H_g(t)$$
 (2.21)

$$f(t) = \sum_{g=1}^{G} f_g(t)$$
 (2.22)

$$F(t) = \sum_{g=1}^{G} F_g(t)$$
 (2.23)

$$S(t) = \prod_{g=1}^{G} S_g(t).$$
 (2.24)

where the subscript g signifies that the function pertains to the g^{th} risk. Further, $H_g(t) = \int_{u=0}^{t} h_g(u) du$ is the cause-specific cumulative hazard and $F_g(t) = \int_{u=0}^{t} f_g(u) du$.

For semi-parametric regression approaches, there are two prevailing methods to construct models that conditionally explain survival in the presence of competing risks. The first focusses on modelling the cause-specific hazard (Pintilie, 2006), which conditionally explains the instantaneous rate of occurrence of the g^{th} risk in subjects who are currently event free – that is, in subjects who have not yet experienced any of the different types of events. This is the method used in chapter 4 of this thesis. The second, introduced by Fine and Gray (1999), focusses on the hazard of the sub-distribution, which conditionally explains the instantaneous risk of failure from the g^{th} event in subjects who have not yet experienced an event of type g, but may or may not have experienced another event. The hazard of the sub-distribution is not explored in this thesis.

Sueyoshi (1992) extends semi-parametric Cox model with competing risks to cater for time-varying covariates, assuming these are piecewise constant (ie: they remain constant between the discrete times they are observed), arguing this is appropriate as there is a lack of a-priori knowledge of the evolution path for their values. Barnett and Graves (2008) discuss time-varying covariates provide richer information, but require non-standard data formatting.

Introducing both baseline and time-covariates, the cause-specific hazard becomes

$$h_{ig}(t) = h_{0g}(t)e^{\mathbf{x}_i^T \mathbf{\beta}_g + \mathbf{z}_i^T(t)\mathbf{\gamma}_g} \quad i = 1, \dots, n \quad and \ g = 1, \dots, G \tag{2.25}$$

where: $h_{0g}(t)$ is the g^{th} baseline hazard; \mathbf{x}_i and $\mathbf{z}_i(t)$ are vectors of baseline and timevarying covariates respectively; and $\boldsymbol{\beta}_g$ and $\boldsymbol{\gamma}_g$ are the accompanying vectors of regression coefficients. Observed data for the i^{th} subject becomes $(y_i, \delta_i, \mathbf{x}_i, \mathbf{z}_i(t))$, so the partial likelihood for (i = 1, ..., n) subjects for all the regression coefficients $\boldsymbol{\beta} = (\boldsymbol{\beta}_1, ..., \boldsymbol{\beta}_G)$ and $\boldsymbol{\gamma} =$ $(\boldsymbol{\gamma}_1,\ldots,\boldsymbol{\gamma}_G)$ is

$$L(\boldsymbol{\beta}) = \prod_{i=1}^{n} \left[\frac{h_{ig}(y_i)}{\sum_{j \in \mathscr{R}_g(\tau_j)} h_j(\tau_j)} \right]^{\mathbf{1}_{\delta_i = g}} = \prod_{i=1}^{n} \prod_{g=1}^{G} \left[\frac{h_{ig}(y_i)}{\sum_{j \in \mathscr{R}_g(\tau_j)} h_j(\tau_j)} \right]^{\mathbf{1}_{\delta_i = g}}.$$
 (2.26)

This is the product of individual partial likelihoods for estimating the g^{th} cause specific hazard but setting all other events to be censored. It thus follows that maximising the g cause-specific hazards individual also maximises equation (2.26). The practicality of this is that to estimate the g^{th} cause-specific hazard, all that is necessary is to set $\delta_i = 0$ for subjects that have an event other than type g. Critically Bakoyannis and Touloumi (2011) discuss that individual estimation of the g cause-specific hazards does not require independence of the competing risks and that while the most common approach is to fit models individually to each event type, there are methods that allow simultaneous estimation.

2.2.3 Mixture-Cure Analysis

Mixture-cure analysis models the time to an event of interest where subjects are from two distinct populations; (1) those susceptible to the event of interest; and (2) those no longer susceptible and hence "cured". These models were initially applied in studies of diseases, beginning with Berkson and Gage (1952) and Farewell (1982) who devised parametric approaches. Later extensions are by authors such as Kuk and Chen (1992), Peng and Dear (2000) and Sy and Taylor (2000) who extend the method to include the semi-parametric Cox model, with Cai et al. (2012) providing an R implementation in their smcure, estimating model parameters using the Expectation-Maximisation approach of Dempster et al. (1977). The implication with the mixture-cure model is that the survival curve for the both the susceptible and non-susceptible populations considered together plateaus at a level equal to the long-run cure probability.

Let R_i be a susceptibility indicator so that $R_i = 1$ if subject *i* is susceptible and $R_i = 0$ for otherwise. Let T_i be the time at which the event of interest occurs for the *i*th subject in the susceptible population, and C_i be the non-informative (independent) right censoring time for the *i*th subject in the susceptible population. So the observable survival time for all subjects is $Y_i = \min(T_i, C_i)$ for susceptible subjects and $Y_i = \infty$ for non-susceptible subjects, with y_i representing observed realisations. Thus each value of y_i can represent the time an event of interest occurs if $\delta_i = 1$, or when $\delta_i = 0$, this individual can be either right censored for the event of interest or belong to the non-susceptible population. To simplify discussions we let the combined survival time be

$$T_i^{\dagger} = \begin{cases} T_i, & \text{if } i^{th} \text{ subject is susceptible} \\ \infty, & \text{if } i^{th} \text{ subject is not susceptible.} \end{cases}$$
(2.27)

In equation (2.28) below we express the relationship between survival functions $S_i^{\dagger}(t) = P(T_i^{\dagger} > t)$ and $S_i(t) = P(T_i > t)$.

For all subjects there are *p* covariates in the vector $\mathbf{x}_i^T = [x_{i1}, \dots, x_{ip}]$ that help explain survival in the susceptible population and *q* covariates in the vector $\mathbf{w}_i^T = [w_{i1}, \dots, w_{iq}]$ that help explain the probability of susceptibility. Defining $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ as regression coefficients and $h_0(t)$ as the non-parametric continuous baseline hazard, the mixture cure model is

$$S_i^{\dagger}(t|\boldsymbol{w}_i, \boldsymbol{x}_i) = \pi_i(\boldsymbol{w}_i)S_i(t|R_i = 1, \boldsymbol{x}_i) + 1 - \pi_i(\boldsymbol{w}_i)$$
(2.28)

where

$$P[R_i = 1] = \pi_i(w_i) = 1/(1 + exp(-w_i^T \alpha))$$
(2.29)

$$h_i(t|R_i = 1, \mathbf{x}_i) = h_0(t)e^{\mathbf{x}_i^T \boldsymbol{\beta}}$$
(2.30)

$$S_i(t|R_i = 1, \boldsymbol{x}_i) = e^{-\int_0^t h_i(\xi|R_i = 1, \boldsymbol{x}_i)d\xi}$$
(2.31)

Note that R_i is unobservable, with instead all that is observable are censored survival times corresponding to either: (1) subjects from the non-susceptible population; or (2) subjects from the susceptible population but whose censoring time occurs prior to their event time.

In equation (2.28) the function $S_i^{\dagger}(t_i|\boldsymbol{w}_i, \boldsymbol{x}_i)$ has the property it "plateaus" at the long-run cure rate $1 - \pi_i(\boldsymbol{w}_i)$ (ie: as $t \to \infty$ then $S_i^{\dagger}(t_i|\boldsymbol{w}_i, \boldsymbol{x}_i) \to 1 - \pi_i(\boldsymbol{w}_i)$). Absent a non-susceptible population (ie: $\pi = 1$), equation (2.28) becomes the survival function $S_i(t|R_i = 1, \boldsymbol{x}_i)$.

2.2.3.1 Estimation via Expectation-Maximisation

Using observed data $(y_i, \delta_i, w_i, x_i)$ (i = 1, ..., n), the observed likelihood is

$$L(\alpha, \beta) = \prod_{i=1}^{n} \left[\pi(w_i) f(y_i | R_i, x_i) \right]^{\delta_i} \left[\left(1 - \pi(w_i) \right) + \pi(w_i) S(y_i | R_i = 1, x_i) \right]^{1 - \delta_i}$$
(2.32)

where $f(y_i|R_i, x_i)$ is the density function for the susceptible population. Note that when all subjects are susceptible (ie: $\pi(w_i) = 1$ for all subjects) then equation (2.32) reduces to the likelihood for the semi-parametric Cox model. Given R_i , the complete likelihood function is

$$L^{Complete}(\boldsymbol{\alpha},\boldsymbol{\beta}) = \prod_{i=1}^{n} \left[\pi(\boldsymbol{w}_{i})^{R_{i}} \left(1 - \pi(\boldsymbol{w}_{i}) \right)^{(1-R_{i})} \right] \times \left[h(y_{i}|R_{i} = 1, \boldsymbol{x}_{i})^{R_{i}\delta_{i}} S(y_{i}|R_{i} = 1, \boldsymbol{x}_{i})^{R_{i}} \right]$$
$$= L^{Incident}(\boldsymbol{\alpha}|R) \times L^{Latent}(\boldsymbol{\beta}|R).$$
(2.33)

The likelihood in equation (2.33) is the product of likelihood functions for a logistic regression model ($L^{Incident}(\alpha|R)$) and a Cox model ($L^{Latent}(\beta|R)$) conditional on susceptible subjects. Equation (2.33) relies on: $(1 - R_i)(1 - \delta_i) = (1 - R_i)$; $\pi(w_i)^{R_i\delta_i}\pi(w_i)^{R_i(1-\delta_i)} = \pi(w_i)^{R_i}$; and $h(y_i|R_i, \mathbf{x}_i) = f(t_i|R_i, \mathbf{x}_i)/S(y_i|R_i, \mathbf{x}_i)$.

The corresponding log-likelihood can be rearranged into a log-likelihood which is the sum of following two components

$$l^{Incident}(\alpha|R) = \sum_{i=1}^{n} \log \left[\pi(w_i)^{R_i} (1 - \pi(w_i))^{R_i} \right]$$
(2.34)

$$l^{Latent}(\boldsymbol{\beta}|\boldsymbol{R}) = \sum_{i=1}^{n} \log \left[h(y_i|R_i = 1, \boldsymbol{x}_i)^{\delta_i R_i} S(y_i|R_i = 1, \boldsymbol{x}_i)^{R_i} \right]$$
(2.35)

It is impossible to segregate all subjects into cured and non-cured individuals at t = 0. This corresponds to the random variable R_i only being observed for subjects whose event time y_i is an event of interest ($\delta_i = 1$ and $R_i = 1$), while random variable R_i is missing for subjects whose event time y_i is censored ($\delta_i = 0$). In this setting, model estimation can instead be carried out using the Expectation-Maximisation algorithm of Dempster et al. (1977). To maximise the likelihood in equation (2.32), an estimate of $E[R_i|Y_i]$ for the expected value of the random variable R_i is needed.

Several authors use the EM algorithm to estimate the mixture-cure model including (for example) Peng and Dear (2000) and Sy and Taylor (2000) with the latter comparing the Breslow estimator for the baseline hazard or the Kaplan-Meir estimator for baseline survival. Yu and Tiwari (2007) use the EM algorithm to fit a mixture-cure model but using additive hazards. Lu (2008) also uses the EM algorithm, but goes on to also estimate a semi-parametric variance bound. Mohammad et al. (2019) adopt a profile likelihood approach to estimate the cumulative baseline hazard and the regression parameters, which "profiles out" the baseline hazard function from the Cox proportional hazard model. The results of this model are compared to the approach implemented in the R package smcure by Chao et al. (2012).

2.2.3.2 Estimation via Other Methods

While many methods in the literature adopt the EM algorithm to estimate the mixture-cure model, there are other published approaches. Examples of parametric approaches include Gu et al. (2011) who develop a proportional odds model with a Weibull baseline hazard to cater for the cure fraction in a cancer study, while Elorant et al. (2014) use relative survival and excess mortality via a parametric mixture-cure model to cancer data. Li et al. (2007) develops a mixture-cure model for dependant censoring using Archimedean copulas. Bayesian examples usually adopt Markov-Chain Monte-Carlo (MCMC) methods, such as Basu and Tiwari (2010) who use MCMC with a marginal likelihood and a piece-wise constant hazard, Coelho et al. (2015) use MCMC Procedure in SAS (SAS (2020)) and a three-parameter Burr distribution for the baseline hazard, while Yin and Ibrahim (2005) use MCMC and a piecewise exponential hazard.

Other published approaches look to parametrise the baseline hazard. Corbiere et al. (2009) estimates the mixture-cure model using penalised maximum likelihood, parametrising the cumulative hazard using cubic I-splines. Hua and Xiang (2013) cater for interval censoring using transformed baseline hazard function using splines. Andersson et al. (2011) fit a flexible parametric cure model which uses log-cumulative excess hazard of the relative risk, fitting their model in Stata with restrictive cubic splines. Patilea and VanKeilegom (2017) model the cure probably parametrically (such as via a logistic transform) but the survival component

of the model non-parametrically, devising an inversion formula based on the cumulative baseline hazard. Liu and Shen (2009) develop a semi-parametric mixture-cure model for interval censored data, parametrising the cumulative baseline hazard using non-decreasing step-functions. Sy and Taylor (2000) employ the EM algorithm to estimate regression parameters and either the Breslow estimator for the baseline hazard or the Kaplan-Meier estimator for baseline survival. Corbiere and Joly (2007) parametrise the baseline survival function using a Weibull and exponential distributions.

Regarding standard error estimation, Sy and Taylor (2000) outline an approach to estimate asymptotic variance of the regression parameters in a mixture-cure model. They apply a zero-tail constraint which sets the baseline survival function for the susceptible population equal to zero at the final observed event. Corbiere and Joly (2007) undertake a non-parametric bootstrap.

This thesis extends the work of Ma et al. (2014) to mixture-cure model with baseline covariates in chapter 5, and investigates results using both simulation and application to modelling the risk of credit default.

2.3 Literature Review - Applications to Credit Risk

Survival analysis was first introduced by Narain (1992) to credit default modelling and now appears widely in the literature as a method to estimate the chance a customer will default (Lessmann et al., 2015). Survival analysis can be used to determine not only if, but when, a customer is likely to default, which is one of the key advantages beyond logistic regression (the most commonly employed method in industry).

Banasik et al. (1999) investigate survival analysis for credit scoring, which is designed to answer the question of how likely an applicant for credit is to default by a given time in the future. The paper compares four models (logistic regression, the Cox model and the Weibull and exponential accelerated lifetime models) finding they all perform similarly given the data in their study. The paper also recognises the competing risk nature of credit risk data, and outlines (but does not fit) an approach that models the risk of credit default as the event of interest and successful repayment as a competing risk. Bellotti and Crook (2009) show that using time-varying macroeconomic variables in the semi-parametric Cox model improves the accuracy of estimating the probability of default compared to both a semi-parametric Cox model (without time-varying covariates) and a logistic regression. The authors also argue that given credit data is typically in a panel format, where new accounts enter, old accounts leave and each account is observed for a sequential period of time, it naturally allows the use of survival models with time-varying covariates. They also state an additional advantage of survival analysis is that it provides probability of default estimates over many different horizons, where logistic regression is restricted to a just single horizon.

Man (2014) explores a probability of default model using consumer and corporate data from Rabobank, developing an algorithm to undertake the binning of covariates using the hazard function (rather than the industry standard weight of evidence (Good, 1950)) as well as devising a method to compare predictions from a survival model and a logistic regression. The results confirms that survival models perform similarly to logistic regression, a finding repeated by other authors (see for example Stepanova and Thomas (2002)). Despite these performance similarities, Man (2014) states that survival models have certain advantages over logistic regression, specifically: (1) less data is discarded because survival analysis can utilise censored observations; and (2) logistic regression only estimates the survival probability for a fixed time interval (for example over one year). Djeundje and Crook (2019) incorporate time-varying coefficients by estimating a discrete-time survival model for credit card data, and parametrise the baseline hazard using B-splines.

Im et al. (2012) develop a semi-parametric Cox model to predict default risk for a sample of United States credit card data. They identify that macroeconomic effects have a marked impact on observed default rates, doubling due to the global financial crisis. Their method does not include macro-economic variables as covariates, instead including indicator variables for each calendar quarter, an effect the authors call a "time-dependency factor". The authors point out that this is not only a function of the macroeconomic effects but also the aggregated effects of all time-dependent factors that are not otherwise accounted for in the remaining predictor variables in the model. Their paper plots the resulting coefficients, which is reproduced in figure 2.4 below. Their model correctly detects the increase in default risk due to the financial crisis in late 2008 and the bursting of the dot-com bubble in early

2004. However the model also detects an apparent increase in the risk of default in the last quarter of 2005, but this was instead subsequently diagnosed by the authors to be the result of a change in collection policy at the bank concerned. The paper concludes with comments that this approach to modelling the macro-economic effects using survival analysis is novel, but would pose substantial challenges if the model were implemented to estimate out-of-time predictions. The authors suggest two alternate methods to counter this shortcoming, both of which use constant extrapolation beyond the in-sample training data.



Figure 2.4: Estimated Coefficients from for US Credit Card Data Model Source: Im et al. (2012) (figure 3)

An important aspect of defaulted loans is that some defaults can "cure", and return to be performing loans and then be again at risk of default. An example of a default that typically cures is forbearance (or hardship), whereby the lender will grant temporary concessionary arrangements, with the expectation that the customer will return to performing – this is an example of recurrent events. Ambrose and Capone (2020) capture the increased hazard of second and subsequent defaults by including a time-varying covariate of "time since last default". They test their approach using USA data from the Federal Housing Administration to develop a Cox proportional hazard model that also considers financial, borrower, and macroeconomic variables.

Mortgage loans face two interdependent competing risks of default and early prepayment, which need simultaneous development and estimation. Hall and Lundstedt (2005) describe why default and prepayment are intertwined, discussing how the all-survival function and the cumulative incidence functions for each competing risk may fluctuate when interest rates rise or when interest rates fall. When rates fall, then prepayments occur faster and defaults occur less frequently. When rates rise, then prepayments occur slower and defaults occur more frequently. The authors also discuss using stylised graphs for the seasoning effects of loans, whereby the hazard for defaults are humped and peak in at about 24 months.

Tong et al. (2012) constructs a mixture-cure model to estimate not only if a customer is likely to default (susceptibility) but when they are likely to default given they are susceptible (survival time). The authors state this type of approach explicitly recognises and caters for the competing risk of successful loan repayment, and has been employed previously to model long-term survival of cancer patients for two distinct subpopulations — those cured who will never relapse; and those uncured who remain susceptible to the disease. The model is trained using data from a United Kingdom personal loan portfolio and the model parameters for the mixture cure model are estimated using the expectation-maximisation (EM) algorithm (Dempster et al., 1977). A 100-fold cross-validation is used to obtain unbiased predictive performance estimates and bootstrapping is used to estimate 95% confidence intervals.

Clapp et al. (2005) compares a multi-nominal logit model with a mixture-cure model. The findings are that the mixture-cure model is superior, but can be difficult in some circumstances for the estimation to converge. The unobserved heterogeneity is captured by including subject-specific random effects, one in each the linear predictors for each competing risk.

Dirick et al. (2015) outlines how mixture cure models are applicable to credit risk modelling as the vast majority of loans do not have an event of interest (default), identifying three susceptible sub-populations: default; maturity; and early redemption. This leads naturally to the overall survival being modelled as a multi-nominal logit model with each event modelled using their own individual conditional survival functions. The model is fit in a benchmark study using ten different real-world data sets from five banks. Dirick et al (2016) extend this work to additionally include macroeconomic variables as time-varying covariates, using a discrete-time model to construct the full likelihood, and uses the EM algorithm to estimate the parameters.

Djeundje and Crook (2018) outline a multi-state model for credit card data and extend the

competing risk survival model to estimate transition intensities between loans that are up-todate, and those that are one, two and three months in arrears. The model estimates each of the 6 models using separate semi-parametric Cox models, parametrising the baseline hazards using B-splines and both baseline and time-varying covariates are included in the model. Random effects cater for dependence between jumps undertaken by the same account. In order to translate these estimates to competing risk probabilities, the authors use an approach common in actuarial mathematics for life contingent risks (see for example Luptakova and Bilikova (2014)).

2.4 Structure of This Thesis

The remainder of this thesis is structured as follows. Chapter 3 is a published paper (Thackham and Ma, 2020a) extending the work of Ma et al. (2014) to estimate a semi-parametric Cox model with time-varying covariates. Chapter 4 is a published paper (Thackham and Ma, 2020b) that further extends Ma et al. (2014) and Thackham and Ma (2020a) to cater for competing risks. Chapter 5 is a manuscript prepared for publication extending Ma et al. (2014) to estimate a semi-parametric mixture-cure Cox model. Each of chapters 3, 4 and 5 outline the necessary theory, undertake simulation studies and apply the methods to estimating credit risk default for home loan data. The thesis ends with a conclusion and discussion. The thesis is supported by an appendix that contains the R-code to fit the models from chapters 3, 4 and 5.

PAPER 1: On Maximum Likelihood Estimation of the Semi-Parametric Cox Model with Time-Varying Covariates

3

This chapter is a paper published in 2020 in The Journal of Applied Statistics.

Thackham, M. and Ma, J. (2020). On maximum likelihood estimation of the semiparametric Cox model with time-varying covariates. Journal of Applied Statistics, 47(9), 1511-1528. DOI: https://doi.org/10.1080/02664763.2019.1681946

3.1 Abstract

Including time-varying covariates is a popular extension to the Cox model and a suitable approach for dealing with non-proportional hazards. However partial likelihood estimation of

this model has three shortcomings: (1) estimated regression coefficients can be less accurate in small samples with heavy censoring; (2) the baseline hazard is not directly estimated; and (3) a covariance matrix for both the regression coefficients and the baseline hazard is not easily produced.

We address these by developing a maximum likelihood approach to jointly estimate regression coefficients and baseline hazard using a constrained optimisation ensuring the latter's non-negativity. We demonstrate asymptotic properties of these estimates and show via simulation their increased accuracy compared to partial likelihood estimates in small samples and show our method produces smoother baseline hazard estimates than the Breslow estimator.

Finally, we apply our method to two examples, including an important real-world financial example to estimate time to default for retail home loans. We demonstrate using our maximum-likelihood estimate for the baseline hazard can give much clearer corroboratory evidence of the "humped hazard", whereby the risk of loan default rises to a peak and then later falls.

Keywords: Cox model, time-varying covariates, maximum likelihood, constrained optimisation.

3.2 Introduction

Survival analysis involves following subjects for an observation period in anticipation an event of interest will occur. If the event does not occur within this observation period, the time to event for this subject is right censored. The Cox model (Cox, 1972) is the corner-stone of modern survival analysis allowing the natural logarithm of the hazard ratio to be a linear function of covariates. It has been applied in areas as diverse as biomedical science, industrial life testing (Kalbfleisch and Prentice, 2002) and finance (Lessmann et al., 2015). Crowley and Hu (1977) extend the model to include time-varying covariates whose values change for a subject while they are in the study. These time-varying covariates can also be used to relax the inherited proportional hazards assumption of the Cox model with time-fixed covariates (Cox and Oakes, 1984).

Time-varying covariates can be either a continuous or a discrete function of time. In this

paper we consider discrete time-varying covariates but cater for the setting where the values of time-varying covariates can change multiple times for a given subject and assume there is no measurement error for both time-fixed and time-varying covariates. An example of such a time-varying covariate in a financial setting is a home loan whose interest rate changes during term of the loan; this can occur never, once, or potentially numerous times.

As is the case for the Cox model with time-fixed covariates, estimation of the regression parameters for time-varying covariates is commonly obtained using the partial likelihood method. In some settings this approach results in:

- 1. potentially inaccurate estimates for coefficients of the time-varying covariates when the sample size is small and with heavy censoring (Heinze and Dunkler, 2008);
- 2. no estimation of the baseline hazard without resort to a separate estimation via the Breslow method.

The full likelihood method we develop in this paper avoids these shortcomings. Estimation of solely regression coefficients is no problem if the analysis aims to draw inferences on covariate effects or calculate hazard ratios. However if the analysis is for inferences on survival probabilities, then an estimate of the baseline hazard is required. The Breslow method provides a secondary estimation of the baseline hazard to which the partial likelihood estimated regression coefficients are required inputs, with resultant baseline hazard estimates typically being highly volatile (e.g. Hosmer et al. (2008)).

While there are likelihood based methods for simultaneous estimation of regression coefficients and baseline hazard with time-fixed covariates (see for example, Cai and Betensky (2003), Joly et al. (1998) and Ma et al. (2014)), similar likelihood-based methods for Cox model with time-varying covariates are not conveniently available. Wong et al. (2017) develop a maximum likelihood approach to estimate a piece-wise (on baseline hazard) proportional hazard model with time-varying covariates, however all the time-varying covariates are piece-wise constant with one or two common change-points.

In this paper, we present a novel maximum-likelihood method that extends the computational approach of Ma et al. (2014) by simultaneous estimation of the regression coefficients and baseline hazard for the Cox model with right censoring when time-varying covariates present. We develop a constrained optimisation algorithm which is efficient and easy to implement and provide an asymptotic covariance matrix for these estimates, allowing inferences to be conducted efficiently on quantities of interest. Our method provides additional features to those devised by Wong et al. (2017), such as: a flexible spline-based approximation to the baseline hazard; proof of consistency and asymptotic normality of the maximum likelihood estimates; and provision of a Hessian matrix for regression and baseline parameters.

This paper proceeds as follows. Section 3.3 introduces the terminology used thoughout our paper before going on to explain our maximum-likelihood method for the Cox model with time-varying covariates. In section 3.4 we develop our constrained optimisation algorithm with section 3.5 studying asymptotic properties of these estimators. Section 3.6 shows results of applying our model in a simulation study and goes on to demonstrate two real-world applications of our methodology. We conclude with a discussion in section 3.7.

3.3 The likelihood function

We begin by outlining the notation used in this paper. Let T_i be the event time for subject iand C_i the corresponding non-informative right censoring time so that the observable survival time is $Y_i = \min(T_i, C_i)$. Denote the observed Y_i by y_i for i = 1, ..., n. Each y_i can be either a time recording an event of interest ($\delta_i = 1$) or censoring time ($\delta_i = 0$). Observations are represented as (y_i, δ_i). We develop a maximum likelihood approach to fit the following semi-parametric Cox model with time-varying covariates:

$$h_i(t) = h_0(t)e^{\mathbf{x}_i^T \boldsymbol{\beta} + \mathbf{z}_i(t)^T \boldsymbol{\gamma}}$$
(3.1)

where β and γ are vectors for regression coefficients, and $h_0(t)$ is the non-parametric baseline hazard function which we assume is continuous. In order for (3.1) to be a valid hazard function it must have $h_0(t) \ge 0$, a constraint we respect in the estimation process we develop in this paper.

The covariates for subject *i* are separated into baseline (time-fixed) and time-varying covariates, so that $\mathbf{x}_i = [x_{i1}, ..., x_{ip}]^T$ is a vector of *p* baseline covariates and $\mathbf{z}_i(t) = [z_{i1}(t), ..., z_{iq}(t)]^T$ is a vector of *q* time-varying covariates.

We re-express (3.1) as

$$h_i(t) = h_{0i}^*(t) e^{\mathbf{x}_i^I \, \boldsymbol{\beta}},\tag{3.2}$$

where $h_{0i}^*(t) = h_0(t)e^{\mathbf{z}_i(t)^T \boldsymbol{\gamma}}$ and estimate simultaneously $\boldsymbol{\beta}, \boldsymbol{\gamma}$ and $h_0(t)$. One noticeable benefit of the simultaneous estimation is that the corresponding asymptotic covariance matrix can be established relatively easily. Estimating $h_0(t)$ without restrictions is infeasible as $h_0(t)$ is an infinite dimensional parameter while we only have a finite number of observations. A common strategy is to simplify $h_0(t)$ to a finite dimensional subspace where its dimension grows with the sample size *n* but at a slower rate. We require that when $n \to \infty$ the simplified $h_0(t)$ converges to the true $h_0(t)$ (Wong and Severini, 1991). The subspace we employ has the dimension $m (\leq n)$ and has non-negative basis functions $\psi_u(t)$ (where $u = 1, \ldots, m$) such that

$$h_0(t) \approx \sum_{u=1}^m \theta_u \psi_u(t), \tag{3.3}$$

where $\psi_u(t) \ge 0$ are non-negative basis functions. We will still denote the right-hand-side of (3.3) by $h_0(t)$ when there is no confusion. While there are many suitable non-negative basis functions for $\psi_u(t)$ our paper focusses has two: (1) indicator functions resulting in a piece-wise constant $h_0(t)$; and (2) M-splines as per Ramsay (1988). The cumulative baseline hazard corresponding to (3.3) is given by $H_0(t) = \sum_{u=1}^m \theta_u \Psi_u(t)$, where $\Psi_u(t) = \int_0^t \psi_u(s) ds$ is the cumulative basis function. Together, this implies that the cumulative hazard for subject *i*, $H_i(t)$, can be written as

$$H_{i}(t) = H_{0i}^{*}(t)e^{\mathbf{x}_{i}^{t}\beta},$$
(3.4)

where $H_{0i}^{*}(t) = \sum_{u} \theta_{u} \Psi_{ui}^{*}(t)$ and

$$\Psi_{ui}^{*}(t) = \int_{0}^{t} \psi_{u}(s) e^{\mathbf{z}_{i}(s)^{T} \boldsymbol{\gamma}} ds.$$
(3.5)

In practical situations, the time-varying covariates $z_{ib}(t)$, b = 1, ..., q, are rarely continuous functions of t. In this paper we consider all $z_{ib}(t)$ are discrete functions, so that $z_{ib}(t)$ remains a constant in each of the n_i time intervals for subject i. For example, when a patient visited a doctor, measurements such as blood pressure, cholesterol level, etc. were taken. These values remained unchanged until the next visit. Thus we assume

$$z_{ib}(t) = z_{iab}I(t_{ia} \le t < t_{i,a+1}), \tag{3.6}$$

where *I* is an indicator function, $a = 1, ..., n_i$ and t_{ia} are called the "changing points" of $z_{ib}(t)$. Without loss of generality we assume $t_{i1} = 0$ and $t_{i,n_i+1} = y_i$. Clearly, each $z_{ib}(t)$ is piece-wise constant with n_i pieces over $[0, y_i]$.

We adopt the "long-format" data frame to accommodate time-varying covariates in our R program. Table 3.1 shows a simple example to elaborate this data frame, where "status" of 0 means no event and 1 means event. Subject 1 has an event time at y_1 and subject 2 has a censoring time at y_2 , and they both have time varying covariates. Subject 1 has three recorded measurements on the time varying covariates and subject 2 has two.

subject	start	end	status	$z_1(t)$	 $z_q(t)$
1	$t_{11} \\ t_{12} \\ t_{13}$	$t_{12} \\ t_{13} \\ y_1$	0 0 1	Z111 Z121 Z131	 Z11q Z12q Z13q
2	$t_{21} \\ t_{22}$	<i>t</i> ₂₂ <i>y</i> ₂	$\begin{array}{c} 0 \\ 0 \end{array}$	Z211 Z221	 Z21q Z22q
3					

Table 3.1: Example of time-varying covariate data frame

For example, Zhang et al. (2018) demonstrate construction of the above data frame can be achieved with the assistance of the survival::tmerge() function from the survival package (Therneau et al., 2015).

Calculation of the integral in (3.5) is simplified when the time varying covariates are discrete. For $t \in (t_{id}, t_{i,d+1}]$ we have

$$\Psi_{ui}^{*}(t) = \sum_{a=1}^{d} [\Psi_{u}(t_{i,a+1}) - \Psi_{u}(t_{ia})] e^{\mathbf{z}_{ia}^{T} \boldsymbol{\gamma}}, \qquad (3.7)$$

where $\mathbf{z}_{ia} = [z_{ia1}, \dots, z_{iaq}]^T$, and therefore

$$H_{0i}^{*}(t) = \sum_{a=1}^{d} [H_0(t_{i,a+1}) - H_0(t_{ia})] e^{\mathbf{z}_{ia}^T \boldsymbol{\gamma}}.$$
(3.8)

Note that $\Psi_u(t_{i1}) = 0$ and $H_0(t_{i1}) = 0$ for all *i* and *u*.

The log-likelihood function from independent survival times y_1, \ldots, y_n is

$$l(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}) = -\sum_{i=1}^{n} H_{0i}^{*}(y_{i}) e^{\mathbf{x}_{i}^{T}\boldsymbol{\beta}} + \sum_{i=1}^{n} \delta_{i}(\log h_{0}(y_{i}) + \mathbf{x}_{i}^{T}\boldsymbol{\beta} + \mathbf{z}_{i,n_{i}+1}^{T}\boldsymbol{\gamma}).$$
(3.9)

Let θ be an *m*-vector for all θ_u . We wish to estimate $\eta = [\beta^T, \gamma^T, \theta^T]^T$ by maximising the log-likelihood subject to the constraints $\theta_u \ge 0$ (since $\psi_u(t) \ge 0$) for u = 1, ..., m. A computational algorithm for this optimisation problem is developed in the next section.

3.4 Constrained optimisation

The Karush–Kuhn–Tucker (KKT) (Karush (1939), Kuhn and Tucker (1951)) first-order necessary conditions for the constrained optimal solution of β , γ and θ are

$$\frac{\partial l}{\partial \beta_j} = 0, \tag{3.10}$$

$$\frac{\partial l}{\partial \gamma_b} = 0, \tag{3.11}$$

$$\frac{\partial l}{\partial \theta_u} = 0 \text{ if } \theta_u > 0 \text{ or } \frac{\partial l}{\partial \theta_u} < 0 \text{ if } \theta_u = 0.$$
 (3.12)

Many available algorithms for constrained optimisation problems such as those given in Luenberger and Ye (2008) are less efficient for our problem when m is large. We therefore instead develop in this section an easy-to-implement algorithm which is efficient for a large m.

We adopt the following strategy to solve equations given in (3.10), (4.16) and (3.12). Beginning with estimates $\beta^{(k)}$, $\gamma^{(k)}$ and $\theta^{(k)}$ at iteration k, iteration k + 1 comprises the following alternating steps.

- 1: Compute $\boldsymbol{\beta}^{(k+1)}$ so that $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\theta}^{(k)}) \ge l(\boldsymbol{\beta}^{(k)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\theta}^{(k)})$.
- 2: Compute $\boldsymbol{\gamma}^{(k+1)}$ so that $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \ge l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\theta}^{(k)}).$
- **3:** Compute $\theta^{(k+1)} \ge 0$ so that $l(\beta^{(k+1)}, \gamma^{(k+1)}, \theta^{(k+1)}) \ge l(\beta^{(k+1)}, \gamma^{(k+1)}, \theta^{(k)})$.

These conditions assure $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\theta}^{(k+1)}) \ge l(\boldsymbol{\beta}^{(k)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\theta}^{(k)})$ at the end of iteration k+1, which is a key requirement for convergence of this algorithm. Steps 1 and 2 are solved by the Newton algorithm incorporating line search steps, and Step 3 is solved by a multiplicative-iterative (MI) algorithm (e.g. Chan and Ma (2012) and Ma (2010)) designed to respect the non-negative constraints on $\boldsymbol{\theta}$. We call this algorithm the Newton-MI algorithm similar to Ma et al. (2014).

To update β , we employ one iteration of the Newton algorithm with line search. Starting with $\beta^{(k)}$ and using a line-search with step size $\omega_1^{(k)} \in (0, 1]$, we have

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \omega_1^{(k)} (\mathbf{X}^T \mathbf{A}^{(k)} \mathbf{X})^{-1} \mathbf{X}^T (-\mathbf{A}^{(k)} \mathbf{1}_n + \boldsymbol{\delta}), \qquad (3.13)$$

where **X** (of size $n \times p$) is the model matrix of time fixed covariates and its *i*-th row is given by \mathbf{x}_i^T , **A** is a diagonal matrix given by $\mathbf{A} = \text{diag}(H_{01}^*(y_1)e^{\mathbf{x}_1^T\boldsymbol{\beta}}, \dots, H_{0n}^*(y_n)e^{\mathbf{x}_n^T\boldsymbol{\beta}})$, $\mathbf{1}_n$ is an *n*-vector of 1's and $\boldsymbol{\delta}$ is an *n*-vector for δ_i 's. Matrix $\mathbf{A}^{(k)}$ is **A** with $\boldsymbol{\beta} = \boldsymbol{\beta}^{(k)}, \boldsymbol{\gamma} = \boldsymbol{\gamma}^{(k)}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$. Matrix $\mathbf{X}^T \mathbf{A} \mathbf{X}$ is the negative Hessian of $l(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\beta}$. The line search parameter $\omega_1^{(k)}$ helps to achieve $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\theta}^{(k)}) \ge l(\boldsymbol{\beta}^{(k)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\theta}^{(k)})$.

Similarly, the Newton method with line search is also applied to update γ . First, let $\mathbf{Z} = [\mathbf{Z}_1^T, \dots, \mathbf{Z}_n^T]^T$, where $\mathbf{Z}_i = [\mathbf{z}_{i1}, \dots, \mathbf{z}_{i,n_i+1}]^T$. Note that \mathbf{z}_{ia} is defined with equation (3.7). Matrix \mathbf{Z} is in fact the model matrix associated with the time-varying covariates, and it has the dimension of $N \times q$, where $N = \sum_i n_i$. Let $\mathbf{B} = \text{diag}(e^{\mathbf{x}_1^T \boldsymbol{\beta}} \mathbf{B}_1, \dots, e^{\mathbf{x}_n^T \boldsymbol{\beta}} \mathbf{B}_n)$, where \mathbf{B}_i is a diagonal matrix of size $n_i \times n_i$ with diagonal elements $[H_0(r_{i,a+1}) - H_0(r_{i,a})]e^{\mathbf{z}_{ia}^T \gamma}$ for $a = 0, \dots, n_i - 1$. Let $\boldsymbol{\zeta}$ be an *N*-vector given by $\boldsymbol{\zeta} = [\zeta_{11}, \dots, \zeta_{1,n_1-1}, \dots, \zeta_{n1}, \dots, \zeta_{n,n_n-1}]^T$, where $\zeta_{ia} = 1$ only if $a = n_i - 1$ and $\delta_i = 1$; otherwise, $\zeta_{ia} = 0$. Then, the Newton algorithm with line search updates γ according to

$$\boldsymbol{\gamma}^{(k+1)} = \boldsymbol{\gamma}^{(k)} + \boldsymbol{\omega}_2^{(k)} (\mathbf{Z}^T \mathbf{B}^{(k)} \mathbf{Z})^{-1} \mathbf{Z}^T (-\mathbf{B}^{(k)} \mathbf{1}_N + \boldsymbol{\zeta}), \qquad (3.14)$$

where $\omega_2^{(k)} \in (0, 1]$ is a line search step size and $\mathbf{B}^{(k)}$ denotes **B** but with $\boldsymbol{\beta} = \boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma} = \boldsymbol{\gamma}^{(k)}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$. Matrix $\mathbf{Z}^T \mathbf{B} \mathbf{Z}$ is the negative Hessian of $l(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\gamma}$. Note that $\omega_2^{(k)}$ is used to achieve $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \ge l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k)}, \boldsymbol{\theta}^{(k)})$.

Finally for updating θ , we adopt the multiplicative-iterative (MI) algorithm which respects

the non-negative constraint on θ and also is very easy to implement. Let both **C** and **C**^{*} be $n \times m$ matrices but their (i, u) elements are $\psi_u(y_i)$ and $\Psi_{ui}^*(y_i)$ respectively. Let δ be the *n*-vector for δ_i and **f** the *n*-vector for $e^{\mathbf{x}_i^T \boldsymbol{\beta}}$. The MI algorithm updates θ according to

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \omega_3^{(k)} \mathbf{S}^{(k)} \left(\mathbf{C}^T [\mathbf{D}^{(k)}]^{-1} \boldsymbol{\delta} - [\mathbf{C}^{*(k)}]^T \mathbf{f}^{(k)} \right),$$
(3.15)

where **D** and **S** are diagonal matrices with elements $h_0(y_i)$ and $\theta_u/(\sum_i \Psi_{ui}^*(y_i)e^{\mathbf{x}_i^T\boldsymbol{\beta}} + \varepsilon)$ respectively, and here ε is a small threshold used to avoid the corresponding denominator being zero. The step size $\omega_3^{(k)} \in (0,1]$ again guarantees that $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\theta}^{(k+1)}) \geq l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\gamma}^{(k+1)}, \boldsymbol{\theta}^{(k)})$.

All the above line searches can be efficiently conducted using, for example, the Armijo rule (Luenberger and Ye, 2008). We comment that the above algorithm is very easy to implement. The updating formula for γ naturally suggests the "long" data format, similar to R "survival" package (Therneau, 2019) for time dependent covariates as demonstrated earlier in Table 4.8. The "start" and "end" times are needed to select the **B** matrix and compute Ψ_{ui}^* values and the ζ vector is determined solely from the "status" column.

When $\mathbf{A}^{1/2}\mathbf{X}$ and $\mathbf{B}^{1/2}\mathbf{Z}$ have full column rank, matrices $\mathbf{X}^T \mathbf{A} \mathbf{X}$ and $\mathbf{Z}^T \mathbf{B} \mathbf{Z}$ are positive definite so that the updates for $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are well defined. Following the same argument as in Chan and Ma (2012) we can show that (i) if $\boldsymbol{\theta}^{(k)}$ is non-negative then $\boldsymbol{\theta}^{(k+1)}$ is also non-negative, and (ii) under certain regularity conditions, this algorithm converges to a solution satisfying the KKT conditions.

3.5 Asymptotic Properties

Asymptotic results are pivotal for performing inferences without relying on computational demanding methods such as bootstrapping which may be less practical for large data sets. Let $(\beta_0, \gamma_0, h_{00}(t))$ be the true parameters. Let $\hat{\beta}$ and $\hat{\gamma}$ be the ML estimates of β and γ respectively. Denote $h_n(t) = \sum_{u=1}^m \theta_u \psi_u(t)$ and $\hat{h}_n(t) = \sum_{u=1}^m \hat{\theta}_u \psi_u(t)$, where $\hat{\theta}_u$ is the ML estimate of θ_u and *m* increases with *n*. Let θ be the vector for all θ_u and $\hat{\theta}$ the vector for $\hat{\theta}_u$.

Following Xu et al. (2018), strong consistency results for $\hat{\beta}$, $\hat{\gamma}$ and $\hat{h}_n(t)$ can be established when $m \to \infty$ and $n \to \infty$ but $m/n \to 0$, with results in Theorem 1 below. These results are mathematically nice but they are practically less useful. Therefore in Theorem 2, we produce a more useful asymptotic normality result for $\hat{\beta}$, $\hat{\gamma}$ and $\hat{\theta}$ for fixed *m*. The simulation results in Section 3.6 demonstrate that the asymptotic variance in Theorem 4 is accurate when compared with the variance obtained from the Monte Carlo method.

The proofs for Theorems 1 and 2 are omitted as they are very similar to the corresponding proofs in Xu et al. (2018). The results in these theorems require some regularity conditions similar to these in Xu et al. (2018). We first state assumptions needed for Theorem 1.

- A1. Matrices **X** and **Z** are bounded and both $E(\mathbf{X}\mathbf{X}^T)$ and $E(\mathbf{Z}\mathbf{Z}^T)$ are non-singular.
- A2. For function $h_n(t)$, its corresponding coefficient vector θ is in a compact subset of \mathbb{R}^m .
- A3. Assume for any $h_0(t)$, there exists a $h_n(t)$ such that $\max_t |h_n(t) h_0(t)| \to 0$ as $m \to \infty$ and $n \to \infty$ but $m/n \to 0$.

We comment that Assumption A3 can be satisfied according to Proposition 2.8 of DeBoor and Daniel (1974).

Theorem 1 Assume Assumptions A1 - A3 hold. Assume there exists an interval [a, b], where $0 \le a < b < \infty$, such that $h_0(t)$ is bounded and has up to $r \ge 1$ derivatives over [a, b]. Assume the number of basis functions satisfy $m = n^{\upsilon}$, where $0 < \upsilon < 1$. Then, when $n \to \infty$,

- 1. $\|\widehat{\boldsymbol{\beta}} \boldsymbol{\beta}_0\| \to 0$ and $\|\widehat{\boldsymbol{\gamma}} \boldsymbol{\gamma}_0\| \to 0$ almost surely, and
- 2. $\sup_{t \in [a,b]} |\widehat{h}_n(t) h_{00}(t)| \to 0$ almost surely.

Next, similar to Yu and Ruppert (2002), we develop a more useful asymptotic normality result for $\hat{\beta}$, $\hat{\gamma}$ and $\hat{\theta}$ assuming a fixed *m*, and therefore *m* is retained in the asymptotic results. The maximum likelihood estimates $\hat{\beta}$, $\hat{\gamma}$ and $\hat{\theta}$ can still achieve \sqrt{n} convergence rate in this context. The fix *m* assumption lies somewhere between parametric and non-parametric modelling as there are freedoms in selecting a value for *m*. A major benefit is that this asymptotic normality can be used to make inferences involving β , γ and $h_0(t)$. This will be helpful in, for example, inferences on survival probabilities or predictions.

Let $\eta = (\theta^T, \beta^T, \gamma^T)^T$, whose length is m + p + q, and it makes the log-likelihood a function of η . The maximum likelihood estimate of η , denoted by $\hat{\eta}$, is obtained by maximizing $l(\boldsymbol{\eta})$ with the constraint $\boldsymbol{\theta} \geq 0$. In developing this asymptotic normality, we must allow the possibility of active constraints (i.e. some $\theta_u = 0$); otherwise, a non-positive definite information matrix can be obtained if some θ_u estimates are 0.

We follow Moore et al. (2008) (Theorem 2) to facilitate the asymptotic normality properties for the constrained maximum likelihood estimates. To elucidate discussions we assume, without loss of generality, that there are *c* active $\theta \ge 0$ constraints in the maximum likelihood estimates. We need to introduce the matrix **U** in Assumption B5 below to indicate the active constraints. Note that $\mathbf{U}^T \mathbf{U} = \mathbf{I}_{(m+p+q-c)\times(m+p+q-c)}$.

The asymptotic normality results for the maximum likelihood estimate $\hat{\eta}$ require the following regularity assumptions.

Assumptions:

- B1. Random vectors $(y_i, \delta_i, \mathbf{x}_i^T, \mathbf{z}_i^T(t))$, i = 1, ..., n, are independent and identically distributed.
- B2. The space Ξ for η is compact.
- B3. $\lim_{n\to\infty} n^{-1}l(\eta)$ exists and has a unique maximum at $\eta_0 \in \Xi$.
- B4. $l(\boldsymbol{\eta})$ is bounded and is twice continuously differentiable in a neighbourhood of $\boldsymbol{\eta}_0$, and the matrix $\mathbf{F}(\boldsymbol{\eta}) = \lim_{n \to \infty} n^{-1} \partial l^2 / \partial \boldsymbol{\eta} \partial \boldsymbol{\eta}^T$ exist.
- B5. Assume there are *c* active constraints from θ . Let $\mathbf{U}_{(m+p+q)\times(m+p+q-c)}$ be a matrix whose rows take zero values corresponding to the active constraints and other rows form an identity matrix. Assume the matrix $\mathbf{U}^T \mathbf{F}(\boldsymbol{\eta}) \mathbf{U}$ is invertible in a neighbourhood of $\boldsymbol{\eta}_0$.

Theorem 2 Assume Assumptions B1 - B5 hold. Assume there are c active constraints in the maximum likelihood estimate of θ and the corresponding U is defined as in Assumption B5. Then, when $n \to \infty$,

- 1. The constrained MPL estimate $\widehat{\eta}$ is consistent for η_0 , and
- 2. $\sqrt{n}(\widehat{\eta} \eta_0)$ converges in distribution to $N(\mathbf{0}, \widetilde{\mathbf{F}}(\eta_0)^{-1})$, where $\widetilde{\mathbf{F}}(\eta)^{-1} = \mathbf{U}(\mathbf{U}^T \mathbf{F}(\eta)\mathbf{U})^{-1}\mathbf{U}^T$.

We comment that matrix $\widetilde{\mathbf{F}}(\boldsymbol{\eta})^{-1}$ is in fact very easy to compute. Firstly, $\mathbf{U}^T \mathbf{F}(\boldsymbol{\eta}) \mathbf{U}$ is obtained simply by deleting the rows and columns of $\mathbf{F}(\boldsymbol{\eta})$ associated with the active constraints. Then $(\mathbf{U}^T \mathbf{F}(\boldsymbol{\eta}) \mathbf{U})^{-1}$ is calculated. Finally, $\widetilde{\mathbf{F}}(\boldsymbol{\eta})^{-1}$ is obtained by padding $(\mathbf{U}^T \mathbf{F}(\boldsymbol{\eta}) \mathbf{U})^{-1}$ with zeros at the positions of the deleted rows and columns. The results in Theorem 4 demand that the active constraints must be identified. This can be easily done as follows. If the maximum likelihood estimates of $\hat{\theta}_u$ is either exactly zero or very close to zero with gradient of less than $(say) - 10^{-2}$ then we treat it as active.

The simulation results reported in Section 3.6 demonstrate that biases in the maximum likelihood estimates are usually negligible and the asymptotic variances of the estimates are accurate.

3.6 Results

In this section we first report the results of a simulation study which compares our maximum likelihood estimates of β , γ and $h_0(t)$ against the partial likelihood estimates where $h_0(t)$ is estimated using the Breslow method for the partial likelihood method. Then we demonstrate our ML estimation method to fit a Cox model with time dependent covariates in two different applied settings: (1) fitting the Stanford Heart Transplant data (Crowley and Hu, 1977); and (2) an important real-world financial example to estimate time to default for retail home loans.

3.6.1 A simulation study

To test the ML method and the algorithm we developed, we simulate independent survival times $y_i = \min(t_i, c_i)$ using the model

$$h_i(t_i) = h_0(t_i)e^{\beta_1 x_{1i} + \beta_2 x_{2i} + \gamma_1 z_{1i}(t_i)},$$
(3.16)

where $h_0(t_i)$ is the baseline hazard. We test two different baseline hazards: (1) of a Weibull distribution: $h_0(t_i) = \lambda^{-\nu} \nu t_i^{\nu-1}$, where $\lambda = 1$ and $\nu = 1.5$; and (2) of an Exponential distribution: $h_0(t_i) = \lambda$ where $\lambda = 1$.

A censoring time c_i , which is independent of t_i , is generated from a uniform distribution

 $U(0, \mu_c)$ for each i, where μ_c is chosen to give, on average, a desired censoring proportion π_c . Thus, independent pairs (y_i, δ_i) for $i = 1, \dots, n$, where δ_i is the event time indicator, are generated. We select the regression coefficients $\beta_1 = \beta_2 = 1$ and $\gamma_1 = 1$ in our simulation. Values for the first baseline covariate x_1 are randomly drawn from a normal N(0,1) distribution and values for the second baseline covariate x_2 are randomly drawn from a Bernouli distribution with parameter 0.5. The time-dependent covariate $z_1(t_i)$ is set to zero at baseline for all subjects. For each subject, its $z_1(t_i)$ changes value three times, each time switching between either zero and one or vice-versa. To solidify, let these three switching times be (t_{1i}, t_{2i}, t_{3i}) such that $0 < t_{1i} < t_{2i} < t_{3i}$, meaning that: for $t \in [0, t_{1i}) z_1(t_i)$ has a value of 0; for $t \in [t_{1i}, t_{2i})$ it has a value of 1; for $t \in [t_{2i}, t_{3i})$ it has a value of 0; and for $t \in [t_{3i}, \infty)$ it has a value of 1. To make the simulation complicated, switching times (t_{1i}, t_{2i}, t_{3i}) are randomly selected individually for each subject, so that no subject has the same set of switching times as any other subject.

We adopt the approach outlined in Austin (2012) to sample survival data with timevarying covariates from a Weibull and exponential distributions. To conduct the Monte Carlo simulation, we draw M = 2000 samples across: two sample sizes (n = 100 and n = 2000); two approximate censoring proportions ($\pi = 20\%$ and $\pi = 80\%$); and two baseline hazards (Exponential and Weibull). The baseline hazards are parametrised using piece-wise constant indicator functions, with knots selected so that there are an equal number of observed events within each interval. The number of piece-wise constant indicator functions are displayed in table 3.2.

For each combination we estimate the baseline and time-dependent effects using our ML method $(\hat{\beta}_1^{ML}, \hat{\beta}_2^{ML}, \hat{\gamma}_1^{ML})$ and compare them with the partial likelihood estimates $(\widehat{\beta}_1^{PL}, \widehat{\beta}_2^{PL}, \widehat{\gamma}_1^{PL})$. Table 3.2 summarises the bias, Monte-Carlo standard deviation (SD) and mean-square error (MSE) for the regression coefficients calculated from the simulation study.

	Exponential Hazard ($\lambda = 1, \nu = 1$)					Weibull Hazard ($\lambda = 1, \nu = 1.5$)				
	Sample	e Size (n)	n=100	n=100	n=2000	n=2000	n=100	n=100	n=2000	n=2000
Ce	nsor Pr	oportion (π)	$\pi = 20\%$	$\pi = 80\%$	$\pi = 20\%$	$\pi = 80\%$	$\pi = 20\%$	$\pi = 80\%$	$\pi = 20\%$	$\pi = 80\%$
No	. Basel	ine Fun. (m)	m=5	m=4	m=16	m=10	m=5	m=4	m=16	m=10
$\widehat{\gamma}_1$	ML	Bias	0.1051	0.0807	0.0016	0.0074	0.1354	0.2998	0.0216	0.0035
		SD	0.4115	0.5584	0.0683	0.1415	0.3998	0.5750	0.1808	0.2647
		MSE	0.1804	0.3184	0.0047	0.0201	0.1782	0.4206	0.0332	0.0701
	PL	Bias	2.2283	4.0046	0.0022	0.0148	3.0103	4.4146	0.0198	0.0029
		SD	5.9563	8.5452	0.2333	0.2884	7.4610	9.3421	0.2570	0.3256
		MSE	40.4435	89.0576	0.0544	0.0834	64.7287	106.7631	0.0664	0.1061
$\widehat{\beta}_1$	ML	Bias	0.0309	0.0361	0.0001	0.0026	-0.0149	0.0369	0.0004	0.0013
		SD	0.1450	0.2051	0.0298	0.0406	0.1387	0.2419	0.0313	0.0483
		MSE	0.0220	0.0434	0.0009	0.0017	0.0194	0.0599	0.0010	0.0023
	PL	Bias	0.0312	0.0290	0.0001	0.0025	0.0230	0.0437	-0.0015	0.0016
		SD	0.1492	0.2052	0.0306	0.0408	0.1512	0.2516	0.0310	0.0482
		MSE	0.0232	0.0430	0.0009	0.0017	0.0234	0.0652	0.0010	0.0023
$\widehat{\beta}_2$	ML	Bias	0.0382	0.0300	0.0033	-0.0035	-0.0126	0.0374	0.0017	-0.0039
		SD	0.2306	0.3475	0.0522	0.0711	0.2284	0.4262	0.0538	0.0872
		MSE	0.0547	0.1216	0.0027	0.0051	0.0523	0.1830	0.0029	0.0076
	PL	Bias	0.0386	0.0229	0.0033	-0.0036	0.0234	0.0424	0.0032	-0.0037
		SD	0.2366	0.3458	0.0527	0.0713	0.2428	0.4345	0.0540	0.0874
		MSE	0.0574	0.1201	0.0028	0.0051	0.0595	0.1905	0.0029	0.0077

Table 3.2: Comparing estimates $\hat{\gamma}_1$, $\hat{\beta}_1$ and $\hat{\beta}_2$ from maximum likelihood (ML) and partial likelihood (PL) methods

These results show that when n = 100 (small sample sizes) the maximum likelihood estimates perform better than partial likelihood on both bias and standard deviations for time-varying covariates. Particularly, in this context, the maximum likelihood method has much smaller biases and standard deviations in recovering the effect for the time-dependent covariate. These conclusions are further substantiated by Figure 3.1 and Figure 3.2, which compare the mean parameter estimates for $\hat{\gamma}_1$ with n = 100.



(b) n=100, 80% censoring, Exponential Hazard

Figure 3.1: Simulation Results for $\hat{\gamma}_1$ (with a censoring proportion of (a) 20%, (b) 80%), comparing maximum likelihood (ML) and partial likelihood (PL) estimation, Exponential Baseline.



(b) n=100, 80% censoring, Weibull Hazard

Figure 3.2: Simulation Results for $\hat{\gamma}_1$ (with a censoring proportion of (a) 20%, (b) 80%), comparing maximum likelihood (ML) and partial likelihood (PL) estimation, Weibull Baseline.

The maximum likelihood estimation of the Cox model also provides an estimate of the baseline hazard. The plots in Figure 3.3 and Figure 3.4 compare the true baseline hazard to the results from each of the four simulations, together with simultaneous 95% confidence intervals (CIs) (where standard deviations were obtained from the Monte Carlo results). The results demonstrate that the baseline hazard can be recovered to a close degree of accuracy by our method.



Figure 3.3: Comparison of true and estimated Exponential baseline hazards, together with simultaneous 95% CIs.





(c) n=100, 20% censoring, Weibull Hazard



(d) n=100, 80% censoring, Weibull Hazard

Figure 3.4: Comparison of true and estimated Weibull baseline hazards, together with simultaneous 95% CIs.

3.6.2 Application 1 - Stanford Heart Data

For our first example we apply our maximum likelihood estimation to the Stanford Heart Transplant data. This dataset has been studied several times, notably by Crowley and Hu (1977) who study the effects of a number of covariates as to whether heart transplant prolongs survival for patients with heart disease. The data follows 103 patients admitted to the Stanford Heart Transplant Program during which time a search for a suitable donor heart is undertaken. This search takes from between a few days to up to a year. Seventy-five patients undergo a transplant, however some patients die prior to a suitable donor heart is found.

Each patient has recorded the date of acceptance T_1 , and the date when they are last observed T_2 , so that their survival time is $T_2 - T_1$ days. If T_2 is the time of death then the

subject's event time is fully observed; if T_2 is the closing date of the study then the subject's event time is right censored. To facilitate modelling with time-varying covariates, we adopt the "long" format where each record has one or several "start" and "stop" times depending on the time-varying covariates. For our application, we select three baseline covariates: "age" – age in years at acceptance minus 45; "year" – acceptance time point after the study began; "surgery" – an indicator if the patient has previously had bypass surgery. A single time-dependent covariate called "transplant" is created to indicate whether the patient has a heart transplant at a particular time. Using this data, we estimate the model

$$h_i(t) = h_0(t)e^{\beta_{age}x_{age} + \beta_{year}x_{year} + \beta_{surg}x_{surg} + \gamma_{tran^{\mathcal{I}}tran^{(t)}}}$$
(3.17)

where: x_{age} is patient age in years at acceptance minus 45; x_{year} is the year of patient acceptance; x_{surg} is an indicator if the patient has previously had bypass surgery (1=yes, 0=no); and $z_{trans}(t)$ is a time-dependent covariate whose values changes from 0 prior to a patient reviving a transplant to 1 post a patient receiving a transplant. The regression parameters ($\beta_{age}, \beta_{year}, \beta_{surg}, \gamma_{trans}$) are unknown and require estimation. For maximum likelihood estimation, we approximate the baseline hazard $h_0(t)$ by 6 cubic M-spline basis functions with knots selected at quantiles of the observed data.

The results of our maximum likelihood and the partial likelihood estimation are detailed in Table 3.3. Each method returns similar accuracy for the regression coefficients and associated standard errors however the maximum likelihood estimation also returns a further 6 parameter estimates for the baseline hazard function displayed in Figure 3.5. In order to compare the

	Maximum Likelihood Estimation				Partial Likelihood Estimation			
Parameter	Value	SE	Z	pval	Value	SE	Z	pval
$\widehat{\beta}_{age}$	0.031	0.014	2.232	0.020	0.027	0.014	1.981	0.024
$\widehat{\beta}_{surg}$	-0.676	0.367	-1.847	0.045	-0.636	0.367	-1.731	0.042
$\widehat{\beta}_{year}$	-0.146	0.070	-2.074	0.021	-0.146	0.070	-2.074	0.019
$\widehat{\gamma}_{tran}$	-0.014	0.312	-0.044	0.482	-0.014	0.313	-0.046	0.482

Table 3.3: Parameter estimates for the Stanford Heart Transplant Data

baseline hazards from the maximum likelihood and partial likelihood methods, we calculate the Breslow estimator for the baseline hazard using the partial likelihood outputs. We plot the results in Figure 3.5.



(a) ML baseline hazard estimate



(b) Breslow baseline hazard estimate

Figure 3.5: Comparison of the baseline hazard estimation from the maximum likelihood (a) and the Breslow (b) methods.

The maximum likelihood estimate of the baseline hazard is clearly smoother than the Breslow estimator, and therefore contains more useful and interpretable information.

3.6.3 Application 2 - Credit Risk Data

In this second example we apply our maximum likelihood estimation to a real-world credit risk dataset comprising a randomised and anonymised sample of n = 100,000 Australian home loans spanning the calendar years 2003 to 2014 and containing the 11 baseline covariates and 2 time-varying covariates as per table 3.4. Note that variable 12 (Dynamic Loan to Value Ratio) and variable 13 (Worst Delinquency in Last 6 Months) are the time-varying covariates. The baseline hazard is parametrised using 11 piece-wise constant indicator functions, with knots selected so that there are an equal number of observed events within each piece-wise constant.

PAPER 1: On Maximum Likelihood Estimation of the Semi-Parametric Cox Model with Time-Varying Covariates

	Variable	Description (* = mean corrected)
1	Mortgage Insured	Does the home loan have lenders mortgagee insurance? (Yes / No)?
2	Borrower's Occupation	Professional / Trades / Sales / Other
3	Borrowers	How many borrowers are there for this home loan (either 1 or 2+)?
4	Salary Credits	Do the borrowers deposit salary into the home loan account (Yes/No)?
5	Credit Card	Do the borrowers have a credit card? (Yes/No)
6	Personal Loan	Do the borrowers have a personal loan? (Yes/No)
7	Repayment Method	Are repayments principal and interest (P&I) or interest only (IO)?
8	Repayment Frequency	Are repayments made monthly or fortnightly/other
9	Borrower Tenure*	How many months has the borrower been a customer $(0,1,2)$?
10	Total Home Loans*	Total number of home loans the borrowers have (integer 1, 2, \dots)
11	Opening Balance*	The original balance when the home loan opened (\$millions)
12	Dynamic Loan to	The current loan balance divided by the current estimate of the value of
	Value Ratio*	the home securing the loan. This is a time-varying covariate, as both the
		numerator and denominator of this ratio vary over time.
13	Worst Delinguency in	The highest number of missed monthly payments within the last 6 months,
	Last 6 Months	lagged by a period of 12 months. This variable can take integer values of
		0, 1, 2,
		(the number of repayments behind a customer is) definitionally begins at
		zero at origination but can change to values greater than zero over time.

Table 3.4: Description of the 11 baseline and 2 time-varying covariates in the Home Loan dataset

		Partial Likelihood		Maximum Likelihood	
Name	Level	Estimate	Std Err	Estimate	Std Err
Mortgage Insured Borrower's Occupation	Yes Profession Trades Other Sales	$\begin{array}{c} 0.1615 \\ -0.5542 \\ 0.0036 \\ -0.3496 \\ 0 \end{array}$	0.0557 0.0647 0.0595 0.0953	$\begin{array}{c} 0.1599 \\ -0.5543 \\ 0.0034 \\ -0.3478 \\ 0 \end{array}$	0.0556 0.0647 0.0595 0.0953
Borrowers Salary Credits Credit Card Personal Loan Repayment Method Repayment Frequency Customer Tenure (Months) Total Home Loans (Count) Opening Balance (\$millions) Dynamic Loan to Value Ratio Worst Delinguency in Last 6 Months	2+ Yes Yes IO Monthly (t) (t-12)	-0.3184 -0.3240 -0.2775 0.2666 0.1720 0.1728 -0.0021 -0.1112 0.5131 2.7924 2.9979	$\begin{array}{c} 0.0484\\ 0.2123\\ 0.0515\\ 0.1019\\ 0.0522\\ 0.0600\\ 0.0003\\ 0.0186\\ 0.1183\\ 0.1303\\ 0.06328\\ \end{array}$	-0.3183 -0.3225 -0.2778 0.2670 0.1729 0.1715 -0.0021 -0.1111 0.5124 2.7917 2.9983	0.0484 0.2123 0.0515 0.1019 0.0522 0.0560 0.0003 0.0186 0.1183 0.1303 0.0632

Table 3.5: Comparison of Parameter Estimates of the Eleven Baseline and Two Time-Varying Covariates Using Maximum Likelihood and Partial Likelihood Estimation

Table 3.5 demonstrates the estimated regression coefficients and standard errors from maximum likelihood and partial likelihood are very similar. The numerical estimates for the baseline hazard are graphed figure 3.6. In order to compare the baseline hazard estimates, we use the Breslow (1972) estimator for the partial likelihood method. Figure 3.6 compares results of the Breslow estimator (grey line) as per the R package survival (Therneau, 2019); overlaid is the baseline estimate from the maximum likelihood method (black line) along with the associate 95% simultaneous confidence intervals (grey shaded area).

56



Figure 3.6: Comparison of Baseline Hazard Using: "Breslow + Partial Likelihood (PL)" verses "Maximum Likelihood (ML)" Estimation

While in general the two estimates tend to follow each other, it is very clear that the Breslow baseline hazard estimate exhibits a substantially larger degree of volatility than the estimate from the maximum likelihood method. There is a clear evidence in the data of the "humped hazard", with the intensity of default risk beginning low before peaking between 24 and 36 months since the loans were originated. This feature is commonly known for credit portfolio in the finance industry (see for example Im et al. (2012) and Bellotti and Crook (2013)).

3.7 Conclusion

Extension of Cox model to time-varying covariates provides not only additional flexibility to explain the time to event data but also the ability to cater for violation in the proportional hazards assumption. The prevailing estimation technique maximises the partial likelihood resulting in estimates for the regression coefficients but not the baseline hazard. This allows the estimation of hazard ratios as well as inferences on the estimated effects. However, in order to recover some other quantities, such as survival probabilities for individual subjects, an estimate of the baseline hazard is needed. Commonly this relies on the Breslow method (Breslow, 1972), which can result in volatile baseline hazard estimates without providing standard errors.

The partial likelihood method for the Cox model with time-varying covariates has three shortcomings that are not well addressed in the literature: (1) the regression estimates can be inaccurate in samples of moderate size and heavy censoring; (2) the baseline hazard is estimated separately to the regression coefficients; and (3) this separate estimation means a covariance matrix for both the estimated regression coefficients and baseline hazard is not conveniently produced.

In this paper we have developed a novel maximum likelihood approach using constrained optimisation. Our method estimates simultaneously regression coefficients and the baseline hazard. We also have developed the asymptomatic properties of our estimates and have implemented our approach in the R programming language.

Using this R implementation, we show via simulations that our maximum likelihood method produces more accurate estimates in small samples with heavy censoring than estimates obtained via partial likelihood. We also show that we can recover the baseline hazard with a suitable degree of accuracy.

3.8 Appendix: Hessian matrix

The second derivatives with respect to β and γ are:

$$\frac{\partial^2 l}{\partial \boldsymbol{\beta} \partial \boldsymbol{\beta}^T} = -\mathbf{X}^T \mathbf{A} \mathbf{X}, \quad \frac{\partial^2 l}{\partial \boldsymbol{\gamma} \partial \boldsymbol{\gamma}^T} = -\mathbf{Z}^T \mathbf{B} \mathbf{Z}.$$

The other second derivatives are:

$$\frac{\partial^2 l}{\partial \theta_u \partial \theta_v} = -\sum_{i=1}^n \delta_i \frac{\psi_u(y_i)\psi_v(y_i)}{h_0(y_i)^2}, \quad \frac{\partial^2 l}{\partial \beta_j \partial \theta_u} = -\sum_{i=1}^n \Psi_{ui}^*(y_i)e^{\mathbf{x}_i^T \boldsymbol{\beta}} x_{ij},$$
$$\frac{\partial^2 l}{\partial \gamma_b \partial \theta_u} = -\sum_{i=1}^n \Psi_{uib}^{**}(y_i)e^{\mathbf{x}_i^T \boldsymbol{\beta}} x_{ij}, \quad \frac{\partial^2 l}{\partial \beta_j \partial \gamma_b} = -\sum_{i=1}^n H_{oib}^{**}(y_i)e^{\mathbf{x}_i^T \boldsymbol{\beta}} x_{ij}.$$
where, if $t \in \mathcal{B}_{id}$,

$$\Psi_{uib}^{**}(t) = \sum_{a=1}^{d} [\Psi_u(r_{ia}) - \Psi_u(r_{i,a-1})] e^{\mathbf{z}_{ia}^T \boldsymbol{\gamma}} z_{iab}$$
$$H_{0ib}^{**}(t) = \sum_{a=1}^{d} [H_0(r_{ia}) - H_0(r_{i,a-1})] e^{\mathbf{z}_{ia}^T \boldsymbol{\gamma}} z_{iab}$$

References

- Andersen, P. K. and Ørnulf, B. and Richard, D. G. and Niels, K. (1993). *Statistical models based on counting processes*. Springer New York.
- Austin, P. (2012). Generating survival times to simulate cox proportional hazards models with time-varying covariates. *Statistics in Medicine*, 31(29), 3946-3958. https://doi.org/10.1002/sim.5452
- Bellotti, T. and Crook, J. (2009). Credit scoring with macroeconomic variables using survival analysis. *Journal of the Operational Research Society*, 60(12), 1699-1707. https://doi.org/10.1057/jors.2008.130
- Bellotti, T., and Crook, J. (2013). Forecasting and stress testing credit card default using dynamic models. International Journal of Forecasting, 29(4), 563-574. https://doi.org/10.1016/j.ijforecast.2013.04.003
- BIS (2006). International Convergence of Capital Measurement and Capital Standards A Revised Framework Comprehensive Version *Bank for International Settlements*. Basel Framework
- Breslow, N. E. (1972). Contribution to the discussion of paper by D.R. Cox. *Journal of the Royal Statistical Society: Series B*, 34, 216–217 https://www.jstor.org/stable/2985181
- Cai, T. and Betensky, R. A. (2003). Hazard regression for interval-censored data with penalized spline. *Biometrics*, 59(3), 570-579. http://www.jstor.org/stable/3695433
- Chan, R. H. and Ma, J. (2012). A multiplicative iterative algorithm for box-constrained penalized likelihood image restoration. *IEEE Trans. Image Processing*, 21, 3168-3181. https://doi.org/10.1109/TIP.2012.2188811

- Cox, D. (1972). Regression models and life tables. *Journal of the Royal Statistical Society, Series B*, 34(2), 187-220. https://doi.org/10.1111/j.2517-6161.1972.tb00899.x
- Cox, D. (1975). Partial likelihood. *Biometrika*, 62(2), 269-276. https://doi.org/10.1093/biomet/62.2.269
- Cox, D. R. and Oakes, D. (1984). *Analysis of Survival Data*. Chapman and Hall, New York, NY.
- Crowley, J. and Hu, M. (1977). Covariance analysis of heart transplant survival data. *Journal of the American Statistical Association*, 72(357), 27-36. https://doi.org/10.1080/01621459.1977.10479903
- DeBoor, C. and Daniel, J. W. (1974). Splines with nonnegative B-spline coefficients. *Mathematics of Computation*, 28(126), 565-568. https://www.jstor.org/stable/2005928
- Dirick, L. Bellotti, T., Claeskens, G., Baesens, B. (2016). Macro-economic factors in credit risk calculations: Including time-varying covariates in mixture cure models. *Journal of Business and Economic Statistics*, 37(1), 40-53. https://doi.org/10.1080/07350015.2016.1260471
- Heinze, G. and Dunkler, D. (2008). Avoiding infinite estimates of time-dependent effects in small-sample survival studies. *Statistics in Medicine*, 27, 6455 6469. https://doi.org/10.1002/sim.3418
- Honore, B. E. and Powell, J. L. (1994). Pairwise difference estimators of censored and truncated regression models. *Journal of Econometrics*, 64(1-2), 241-278. https://doi.org/10.1016/0304-4076(94)90065-5
- Hosmer, D., Lemeshow, S., and May, S. (2008). Applied Survival Analysis: Regression Modeling of Time to Event Data, second ed. Wiley-Interscience, Hoboken, New Jersey.
- IASB. (2014). International Financial Reporting Standard 9. https://www.ifrs.org/issuedstandards/list-of-standards/ifrs-9-financial-instruments/
- Im, J., Apley, D., Qi, C., Shan, X. (2012) A time-dependent proportional hazards survival model for credit risk analysis. *Journal of the Operational Research Society*, 63(3), 306-321. https://doi.org/10.1057/jors.2011.34

- Joly, P., Commenges, D., and Letenneur, L. (1998). A penalized likelihood approach for arbitrarily censored and truncated data: Application to age-specific incidence of dementia. *Biometrics*, 54(1), 185-194. https://pubmed.ncbi.nlm.nih.gov/9574965/
- Kalbfleisch, J. D. and Prentice, R. L. (2002). *The Statistical Analysis of Failure Time Data*. John Wiley and Sons, New York, NY.
- Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. [Master's thesis, M.Sc. Dissertation. Dept. of Mathematics], Univ. of Chicago.
- Kuhn, H. and Tucker (1951). Nonlinear programming. *Proceedings of 2nd Berkeley Symposium. Berkeley*, pp481-492.
- Lessmann, S., Baesens, B., Seowd, H., and Thomas, L. (2015). Benchmarking state-of-theart classification algorithms for credit scoring: An update of research. *The Journal of the Operational Research Society*, 247(1), 124-136. https://doi.org/10.1016/j.ejor.2015.05.030
- Luenberger, D. and Ye, Y. (2008). *Linear and Nonlinear Programming, 3 ed.* Springer, New York, NY.
- Ma, J. (2010). Positively constrained multiplicative iterative algorithm for maximum penalized likelihood tomographic reconstruction. *IEEE Transactions on Nuclear Science*, 57(1), 181-192. https://doi.org/10.1109/TNS.2009.2034462
- Ma, J., Heritier, S., and Lô, S. (2014). On the maximum penalized likelihood approach for proportional hazard models with right censored survival data. *Computational Statistics and Data Analysis*, 74, 142-156. https://doi.org/10.1016/j.csda.2014.01.005
- Moore, T. J., Sadler, B. M. and Kozick, R. J. (2008). Maximum-likelihood estimation, the Cramer-Rao bound, and the method of scoring with parameter constraints. *IEEE Transactions of Signal Process*, 56, 895-908. https://ieeexplore.ieee.org/document/4451293
- Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science*, 3(4), 425-441. https://doi.org/10.1214/ss/1177012761
- R Core Team. (2020). *R* 4.0.0: A Language and Environment for Statistical Computing. *R* Foundation for Statistical Computing. http://www.R-project.org

SAS Institute Inc. (2020). SAS Software, Version 9.4. http://www.sas.com

- Therneau, T. M. (2019). *A Package for Survival Analysis in R*. (R package version 3.1-6). https://CRAN.R-project.org/package=survival
- Therneau, T., Crowson, C., and Atkinson, E. (2015). Using time dependent covariates and time dependent coefficients in the Cox model. R CRAN Vignette. https://cran.rproject.org/web/packages/survival/vignettes/timedep.pdf
- Tong, T., Mues, C., and Thomas, L. (2012). Mixture cure models in credit scoring: If and when borrowers default. *European Journal of Operational Research*, 218(1). 132-139. https://doi.org/10.1016/j.ejor.2011.10.007
- Wong, W. H. and Severini, T. A. (1991). On maximum likelihood estimation in infinite dimensional parameter spaces. Ann. Statist., 19(2), 603-632. https://doi.org/10.1214/aos/1176348113
- Wong, G. Y. C., Osborne, M. P., Diao, Q., Yu, Q. (2017). Piecewise Cox models with rightcensored data. Communications in Statistics - Simulation and Computation, 46()10), 7894-7908. https://doi.org/10.1080/03610918.2016.1255968
- Xu, J., Ma, J., Connors, M. H. and Brodaty, H. (2018). Proportional hazard model estimation under dependent censoring using copulas and penalized likelihood. *Statistics in Medicine*, 37, 2238-2251. https://doi.org/10.1002/sim.7651
- Yu, Y. and Ruppert, D. (2002). Penalized spline estimation for partially linear single index models. J. Amer. Statist. Assoc., 97(460), 1042-1054. https://doi.org/10.1198/016214502388618861
- Zangwill, W. (1969). *Nonlinear Programming: A Unified Approach*. Prentice-Hall, New Jersey.
- Zhang Z., Reinikainen J., Adeleke K.A., Pieterse M.E., Groothuis-Oudshoorn C.G. (2018). *Time-varying covariates and coefficients in Cox regression models*. *Annals of Translational Medicine*, 6(7), 121. https://doi.org/10.21037/atm.2018.02.12

Zheng, D. and Lin, D. (2007). Maximum likelihood estimation in semiparametric regression models with censored data. *Journal of the Royal Statistical Society B*, 69, 507-564. https://doi.org/10.1111/j.1369-7412.2007.00606.x

4

PAPER 2: On Maximum Likelihood Estimation of Competing Risks using the Cause-Specific Semi-Parametric Cox Model with Time-Varying Covariates – an Application to Credit Risk

This chapter is a paper published in 2020 in The Journal of the Operational Research Society.

Thackham, M. and Ma, J. (2020). On maximum likelihood estimation of competing risks using the cause-specific semi-parametric Cox model with time-varying covariates – an application to credit risk. Accepted in July 2020. DOI: https://doi.org/10.1080/01605682. 2020.1800418

4.1 Abstract

Credit-granting institutions need to estimate the probability of loan default, which represents the chance a customer fails to make repayments as promised. Critically this estimation is intertwined with the competing risk a customer fully repays their loan while also having key predictive drivers with values that change over time. A conventional model in this setting is a competing risks Cox Model with time-varying covariates. However partial likelihood estimation of this model has two shortcomings: (1) the baseline hazard is not estimated, so calculating probabilities requires a further estimation step; and (2) a covariance matrix for both regression coefficients and the baseline hazard is not produced.

This paper caters for these shortcomings by devising a maximum likelihood technique to jointly estimate regression coefficients and the cause-specific baseline hazards using constrained optimisation to ensure the latter's non-negativity. We show via simulation our technique produces regression coefficients estimates with lower bias in small samples with heavy censoring. When applied to a real-world credit risk dataset consisting of home loan data our Maximum Likelihood approach produces a smoother estimate of the cause-specific baseline hazards for default and redemption than those obtained using the Partial Likelihood and Breslow approach. This provides better clarity of the shape of these functions through both a less volatile central estimate as well as quantifying the error of this central estimate. We implement our method in R.

Keywords: Competing Risk Cause Specific Cox Models, Time-varying Covariates, Constrained Maximum Likelihood Optimisation, Credit Risk.

4.2 Introduction

Credit-granting institutions are in the business of lending money to customers, some of whom may fail to repay in a timely manner monies they contractually owe (namely principal, interest and fees), thereby defaulting on their obligation. An accurate estimate of the probability of default is vital for two reasons. The first is for institutions granted permission to use the Internal Ratings-Based approach by their regulator to calculate minimum regulatory credit capital (BIS, 2006). The second is for institutions to satisfy international accounting standards to calculate expected credit losses (IASB (2014), FASB (2016)). However loans

66

that repay monies owed earlier than contractually due can impact the probability of loan default, with Wycinka (2019) outlining competing risks models can appropriately cater for this.

Competing risks models analyse time to event of interest where there are two or more possible events subjects can encounter. "Competing" means events for each subject are mutually exclusive so the occurrence of an event precludes any other event from happening. For example if the event of interest is home loan default then a competing risk is early repayment. Often competing events are ignored and studies focus on the primary event by simply treating competing events as censored (Austin et al., 2016). This can lead to biased estimates; for example Putter et al. (2007) demonstrates when applying the Kaplan-Meier (1958) estimate to the event of interest while treating all other competing events as censored under estimates the survival function of the event of interest.

There are two prevailing regression models for competing risks. The first is the causespecific hazard Cox (1972, 1975) model of Prentice et al. (1978) and the second is the sub-distribution hazard Cox model of Fine and Gray (1999). Our paper focuses on the cause-specific Cox model which estimates the instantaneous rate of a risk for subjects who are currently risk free.

Competing risks analysis have been applied to credit risk (see Banasik et al. (1999) and Stepanova and Thomas (2002)). Deng et al. (2000) estimate cause-specific hazard models for early repayment and default for commercial mortgages. Wycinka (2019) demonstrates when managing a credit portfolio, early repayments alter the probability of loan default. Time-varying covariates (Crowley and Hu, 1977) are often included in the cause-specific Cox regression models. Bellotti and Crook (2009) show that using time-varying macroeconomic variables improves the predictive accuracy, and when analysing a sample of United States credit card data, Im et al. (2012) show that macroeconomic effects have a marked impact on observed default rates which doubled due to the global financial crisis.

Estimating competing risks cause-specific hazards Cox models with time-varying covariates is most commonly undertaken by maximising the partial likelihood (see for example, Pintilie (2006)). However, the partial likelihood approach suffers from the following shortcomings:

1. it does not directly produce an estimation of the baseline hazards. A common remedy is

to adopt a separate estimation via the Breslow method (1972), which typically returns baseline hazard estimates that are highly volatile (e.g. Hosmer et al. (2008))

- it does not provide a covariance matrix for the regression coefficients and the baseline hazard, so joint inferences cannot be formed
- 3. it may give potentially inaccurate estimates for coefficients of the time-varying covariates when the sample size is small and with heavy censoring.

We develop an approach that uses the full likelihood thus avoiding these shortcomings, estimating regression parameters as well as simultaneously providing estimates of the baseline cause-specific hazards. The latter is necessary for drawing inferences on, for example, the survival function. The method devises a novel maximum likelihood method that extends the computational approach of Thackham and Ma (2020), developing a constrained optimisation algorithm which is efficient and easy to implement – which we do so in R – and produces an asymptotic covariance matrix for all the model parameters, allowing inferences to be conducted efficiently on quantities of interest.

This paper proceeds as follows. Section 4.3 devises the log-likelihood function for the competing risk Cox model with time-varying covariates and uninformative right-censoring. Section 4.4 details simulation results comparing our method with the conventional partial likelihood method, going on to apply our method in a real credit setting generating insights to a real-world credit risk dataset. Our paper concludes with discussions in Section 4.5. The accompanying supplementary material details a special constrained optimisation algorithm for computing the maximum likelihood estimates of regression coefficients and baseline hazards, along with asymptotic results.

4.3 The likelihood function

For subject *i*, *i* = 1,...,*n*, let T_i be the true survival time (which may not be observed due to censoring), C_i the corresponding non-informative right-censoring time, thus observed survival time is $Y_i = \min(T_i, C_i)$. Each T_i is associated with g = 1, ..., G mutually excluding competing risks. Realised values for random variable Y_i are y_i and the tuple of observed values for subject *i* is $(y_i, \delta_{i1}, ..., \delta_{iG})$ where $\delta_{ig} = 1$ when subject *i* encounters risk *g*. Additionally we define $\delta_{i0} = \sum_{g=1}^{G} \delta_{ig}$ so $\delta_{i0} = 0$ when the event time for subject *i* is censored. We develop a maximum likelihood approach to fit the following Cox model for each of g, g = 1, ..., G, competing risks cause specific hazard model with time-varying covariates:

$$h_{ig}(t) = h_{0g}(t)e^{\mathbf{x}_i^I \boldsymbol{\beta}_g + \mathbf{z}_i^I(t)\boldsymbol{\gamma}_g}$$
(4.1)

where $h_{ig}(t)$ denotes the hazard of subject *i* for risk *g*, β_g and γ_g are vectors respectively for regression coefficients for time-fixed and time-varying covariates, and $h_{0g}(t)$ the nonparametric baseline hazard function. For (4.1) to be a valid hazard function it must have $h_{0g}(t) \ge 0$ for all *g*, constraints we respect in the estimation process we develop in this paper. Covariates are separated into *p* time-fixed covariates $\mathbf{x}_i = [x_{i1}, ..., x_{ip}]^T$ and *q* time-varying covariates $\mathbf{z}_i(t) = [z_{i1}(t), ..., z_{iq}(t)]^T$.

We re-express (2.2) as

$$h_{ig}(t) = h_{0ig}^*(t)e^{\mathbf{x}_i^T \boldsymbol{\beta}_g} \tag{4.2}$$

where $h_{0ig}^*(t) = h_{0g}(t)e^{\mathbf{z}_i(t)^T \boldsymbol{\gamma}_g}$. Estimating each $h_{0g}(t)$ without restrictions is an ill-conditioned problem. To cater for this we simplify $h_{0g}(t)$ to a finite dimensional subspace where its dimension grows with the sample size n but at a slower rate and further require that when $n \to \infty$ the simplified $h_{0g}(t)$ converges to the true $h_{0g}(t)$ (Wong and Severini, 1991). The subspace we employ to achieve this has the dimension m_g ($\leq n$) with non-negative basis functions $\psi_{gu}(t) \ge 0$ (where $u = 1, \ldots, m_g$) giving

$$h_{0g}(t) \approx \sum_{u=1}^{m_g} \theta_{gu} \psi_{gu}(t).$$
(4.3)

We will still denote the right-hand-side of (4.3) by $h_{0g}(t)$ when there is no confusion.

While there are many suitable non-negative basis functions for $\psi_{gu}(t)$, this paper focusses on an indicator functions resulting in a piece-wise constant $h_{0g}(t)$. Our method can be easily adapted to other non-negative basis functions, for example M-splines as per Ramsay (1988). The cumulative baseline hazard corresponding to (4.3) is given by $H_{0g}(t) = \sum_{u=1}^{m} \theta_{gu} \Psi_{gu}(t)$, where $\Psi_{gu}(t) = \int_{0}^{t} \psi_{gu}(s) ds$ is the cumulative basis function. Together, the cumulative hazard $H_{ig}(t)$ is

$$H_{ig}(t) = H_{0ig}^*(t)e^{\mathbf{x}_i^t \boldsymbol{\beta}_g}, \qquad (4.4)$$

where $H_{0ig}^{*}(t) = \sum_{u} \theta_{gu} \Psi_{gui}^{*}(t)$ and

$$\Psi_{gui}^*(t) = \int_0^t \psi_{gu}(s) e^{\mathbf{z}_i(s)^T \boldsymbol{\gamma}_g} ds.$$
(4.5)

Our algorithm demands evaluation of $\Psi_{gui}^*(t)$ for all y_i within each iteration which is a computational burden. One simplification is to discretise the time-varying covariates. In practice, time-varying covariates $z_{ib}(t)$, b = 1, ..., q rarely appear as a continuous function of time t since its measurements are usually taken over a finite number of time points. For example, the balance of a customer's loan could change several times within each month but for risk modelling purposes end-of-month loan balances may be sufficient for monitoring and modelling the risk of the loan. In this paper we consider all $z_{ib}(t)$ are discrete functions, so $z_{ib}(t)$ remains a constant in each of the time intervals for subject i. Thus

$$z_{ib}(t) = z_{iab}I(t_{ia} \le t < t_{i,a+1}), \tag{4.6}$$

where *I* is an indicator function, $a = 1, ..., n_i$. Points t_{ia} define bins for the discretised "changing points" of $z_{ib}(t)$, forming n_i bins. Change points are common for all time-varying covariates of subject *i*, obtained by combining change points of the time-varying covariates of subject *i*. Without loss of generality we assume $t_{i1} = 0$ and $t_{i,n_i+1} = y_i$.

Calculation of the integral in (4.5) is simplified when time-varying covariates are discrete. For $t \in (t_{id}, t_{i,d+1}]$

$$\Psi_{gui}^{*}(t) = \sum_{a=1}^{d} [\Psi_{gu}(t_{i,a+1}) - \Psi_{gu}(t_{ia})] e^{\mathbf{z}_{ia}^{T} \boldsymbol{\gamma}_{g}}, \qquad (4.7)$$

where $\mathbf{z}_{ia} = [z_{ia1}, \ldots, z_{iaq}]^T$, and therefore

$$H_{0ig}^{*}(t) = \sum_{a=1}^{d} [H_{0g}(t_{i,a+1}) - H_0(t_{ia})] e^{\mathbf{z}_{ia}^{T} \boldsymbol{\gamma}}.$$
(4.8)

Note $\Psi_{gu}(t_{i1}) = 0$ and $H_{0g}(t_{i1}) = 0$ for all *i*, *u* and competing risk *g*.

Let $\boldsymbol{\theta}_g$ be the m_g -vector for all $\boldsymbol{\theta}_{gu} \geq 0$. Let $\boldsymbol{\beta} = [\boldsymbol{\beta}_1^T, \dots, \boldsymbol{\beta}_G^T]^T, \boldsymbol{\gamma} = [\boldsymbol{\gamma}_1^T, \dots, \boldsymbol{\gamma}_G^T]^T$ and $\boldsymbol{\theta} = [\boldsymbol{\theta}_1^T, \dots, \boldsymbol{\theta}_G^T]^T$. The log-likelihood function from independent survival times y_1, \dots, y_n

and competing risks $g = 1, \ldots, G$ is

$$l(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}) = \sum_{g=1}^{G} \left[-\sum_{i=1}^{n} H_{0ig}^{*}(y_{i}) e^{\mathbf{x}_{i}^{T} \boldsymbol{\beta}_{g}} + \sum_{i=1}^{n} \delta_{ig} (\log h_{0g}(y_{i}) + \mathbf{x}_{i}^{T} \boldsymbol{\beta}_{g} + \mathbf{z}_{i,n_{i}+1}^{T} \boldsymbol{\gamma}_{g}) \right].$$
(4.9)

The supplementary material develops our constrained optimisation algorithm which estimates β , γ and θ by maximising (4.9) subject to $\theta \ge 0$.

4.4 Results

4.4.1 Simulation Results

We compare our maximum likelihood method with the conventional partial likelihood method by simulating competing risk data for two competing risks, with the following Weibull baseline hazards:

$$h_{0g}(t) = \lambda_g \nu_g t^{\nu_g - 1}, \ g = 1,2$$
(4.10)

where v_g and λ_g are the shape and scale parameters respectively for risk g. Our simulation sets to $v_1 = v_2 = 1.5$ and $\lambda_1 = \lambda_2 = 1.0$. We consider G = 2 competing risks where each cause-specific hazard includes two baseline covariates x_1 and x_2 and a single time-varying covariate z(t) resulting in

$$h_{ig}(t) = h_{0g}(t) \exp(\beta_{g1} x_{i1} + \beta_{g2} x_{i2} + \gamma_g z_i(t)).$$
(4.11)

We define $z_i(t)$ as a dichotomous time-varying covariate with at most one change from untreated (z = 0) to treated (z = 1), and if subject *i* does undergo this switch, it occurs at time $t = t_{i1}$. For baseline covariates, we select $x_{i1} \sim N(0, 1)$ and $x_{i2} \sim Bernouli(0.5)$. Extending the approach of Austin (2012) from a single risk to two competing risks, the corresponding cumulative hazard is

$$H_{i}(t) = \sum_{g=1}^{2} \lambda_{g} \exp(\beta_{g1} x_{ig} + \beta_{g2} x_{ig}) \left[t_{i1}^{\nu_{g}} + t^{\nu_{1}} \exp(\gamma_{g}) - t_{i1}^{\nu_{1}} \exp(\gamma_{g}) \right].$$
(4.12)

As per Bender et al. (2005), a single survival time *T* from the competing risks model with the hazard in equation (4.11) can be generated by inverting the expression in equation (4.12), such that $T = H^{-1}[-\log(u)]$ where $u \sim Unif(0, 1)$.

To simulate the event, let W_{t_i} be the event that subject *i* encounters, so that $W_{t_i} \in [0, 1, 2]$. All subjects begin at time $t_i = 0$ with $W_0 = 0$ and persist there until transitioning at time $t_i = T_i$ to experience either event 1 ($W_{T_i} = 1$) or event 2 ($W_{T_i} = 2$). As per Beyersmann et al. (2009), the probability that the event which occurs at time $t_i = T_i$ is event 1 is

$$P[W_{T_i} = 1 | t_i < T_i \le t_i + dt, T_i \ge t_i] = \frac{P[t_i < T_i \le t_i + dt, W_{T_i} = 1]}{P[T_i \le t_i + dt | T_i \ge t_i]} = \frac{h_1(t_i)}{h_1(t_i) + h_2(t_i)} = p$$
(4.13)

so that the event of risk 1 occurring is sampled form the Bernoulli distribution Bernoulli(p), where the parameter p is equation (4.13).

Right-censoring times are independently drawn from a uniform distribution parametrised so that two different censoring proportions ($\pi = 10\%$ and $\pi = 40\%$) are realized. The regression coefficients β_{g1} , β_{g2} and γ_g are set to 1.0. Two different sample-sizes are used: n = 1000 and n = 100. We draw M = 1000 replications for each combinations n and π , and then estimate the regression coefficients of the two cause-specific hazards using partial likelihood (PL) and our maximum likelihood (ML) method. Results for the bias, standard deviation and mean-square error of the model parameters in Table 4.1 demonstrate that the ML method can more effectively recover the true parameter value for the time-varying covariate than the PL method in the smaller sample size with heavier censoring. Figures 4.1a and 4.1b further draw this out by displaying the mean and the empirical confidence intervals. Table 4.2 displays the area under the curve (AUC) statistic for each simulation setting along with empirical confidence intervals, showing the ML method has better accuracy in the smaller sample size with heavier censoring. Figure 4.2 chart how the simulated results recover the true Weibull baseline hazard function, with accompanying empirical confidence intervals.

			n = 1000	$\pi = 10\%$	n = 1000	$\pi = 40\%$	n = 100	$\pi = 10\%$	n = 100,	$\pi = 40\%$
Risk	Parameter	Measure	ML	PL	ML	PL	ML	PL	ML	PL
1	\widehat{eta}_{11}	Bias SD MSE	-0.0051 0.0566 0.0032	-0.0150 0.0577 0.0036	-0.0034 0.0669 0.0720	-0.0059 0.0664 0.0717	$0.0350 \\ 0.2045 \\ 0.0430$	$0.0374 \\ 0.2068 \\ 0.0442$	$0.0509 \\ 0.2546 \\ 0.0674$	0.0665 0.2541 0.0690
	\widehat{eta}_{12}	Bias SD MSE	$\begin{array}{c} 0.0044 \\ 0.1031 \\ 0.0106 \end{array}$	$0.0127 \\ 0.1034 \\ 0.0108$	$0.0150 \\ 0.1169 \\ 0.1235$	$0.0161 \\ 0.1172 \\ 0.1237$	$0.0452 \\ 0.3327 \\ 0.1127$	$0.0464 \\ 0.3331 \\ 0.1131$	$0.0715 \\ 0.3993 \\ 0.1645$	$0.0899 \\ 0.4086 \\ 0.1750$
	$\widehat{\gamma}_1$	Bias SD MSE	$0.0075 \\ 0.3706 \\ 0.1374$	$\begin{array}{c} 0.0698 \\ 0.4384 \\ 0.1971 \end{array}$	$\begin{array}{c} 0.0196 \\ 0.1863 \\ 0.1879 \end{array}$	$0.0263 \\ 0.1941 \\ 0.1904$	-0.1253 1.6968 2.8950	5.2011 11.0337 148.7937	$\begin{array}{c} 0.3572 \\ 2.4026 \\ 5.9002 \end{array}$	4.4639 12.2002 168.77
2	\widehat{eta}_{21}	Bias SD MSE	-0.0048 0.0603 0.0037	$0.0048 \\ 0.0605 \\ 0.0037$	$\begin{array}{c} 0.0051 \\ 0.0697 \\ 0.0049 \end{array}$	$\begin{array}{c} 0.0066 \\ 0.0697 \\ 0.0049 \end{array}$	$0.0227 \\ 0.2139 \\ 0.0463$	$0.0295 \\ 0.2177 \\ 0.0483$	$\begin{array}{c} 0.0359 \\ 0.2563 \\ 0.0670 \end{array}$	$0.0526 \\ 0.2584 \\ 0.0695$
	\widehat{eta}_{22}	Bias SD MSE	$\begin{array}{c} 0.0006 \\ 0.1057 \\ 0.0112 \end{array}$	-0.0073 0.1054 0.0112	$0.0004 \\ 0.1232 \\ 0.0152$	-0.0007 0.1234 0.0152	$-0.0056 \\ 0.3289 \\ 0.1082$	-0.0158 0.3261 0.1066	-0.0313 0.4432 0.1974	-0.0401 0.4392 0.1945
	$\widehat{\gamma}_2$	Bias SD MSE	-0.0252 0.4021 0.1623	$0.0483 \\ 0.4285 \\ 0.1859$	$\begin{array}{c} 0.0098 \\ 0.2071 \\ 0.0430 \end{array}$	$0.0175 \\ 0.2167 \\ 0.0473$	$\begin{array}{c} 0.3532 \\ 1.7409 \\ 3.1556 \end{array}$	$\begin{array}{r} 4.4582 \\ 10.5153 \\ 130.446 \end{array}$	0.5264 2.2961 5.5491	4.7883 11.0397 144.8018

Table 4.1: Comparing estimates $\hat{\beta}_{11}$, $\hat{\beta}_{12}$, $\hat{\gamma}_1$, $\hat{\beta}_{21}$, $\hat{\beta}_{22}$, and $\hat{\gamma}_2$, from maximum likelihood (ML) and partial likelihood (PL) methods

RiskMethod	$n = 1000, \pi = 10\%$	$n = 1000, \pi = 40\%$	$n = 100, \pi = 10\%$	$n = 100, \pi = 40\%$
1 ML PL+Bres 2 ML PL+Bres	$\begin{array}{c} 0.7436 \left(0.7164, 0.7719 \right) \\ 0.7423 \left(0.7006, 0.7866 \right) \\ 0.7461 \left(0.7144, 0.7774 \right) \\ 0.7466 \left(0.7001, 0.7945 \right) \end{array}$	$\begin{array}{c} 0.7454 \left(0.7165, 0.7754 \right) \\ 0.7444 \left(0.7114, 0.7842 \right) \\ 0.7488 \left(0.7137, 0.7741 \right) \\ 0.7442 \left(0.7042, 0.7849 \right) \end{array}$	$\begin{array}{c} 0.7577(0.6414,0.8743)\\ 0.7073(0.5252,0.8982)\\ 0.7571(0.6481,0.8840)\\ 0.6995(0.5267,0.9067)\end{array}$	$\begin{array}{c} 0.7589 \ (0.6439, \ 0.8690) \\ 0.6723 \ (0.5120, \ 0.9020) \\ 0.7507 \ (0.6195, \ 0.8804) \\ 0.6813 \ (0.5187, \ 0.9101) \end{array}$

Table 4.2: Comparison of AUC (with 95% CI), from maximum likelihood (ML) and partial likelihood (PL) plus Breslow methods







Figure 4.2: Comparison of true and Maximum Likelihood estimated baseline hazards, together with simultaneous 95% CIs.

4.4.2 Application to Credit Risk Data

We compare our maximum likelihood estimation to the partial likelihood estimation using a real-world credit risk dataset comprising a randomised and anonymised sample of n = 80,000 Australian home loans originated in the calendar years 2003 to 2014. Loans were followed from origination until the earlier of either: loan default; loan redemption; or the end of 2015. The event of interest is default which is defined when the loan has one or more of the following events occur: continuously 90 days past due for an amount greater than \$1000; granting of forbearance; or borrower bankruptcy. The competing risk is redemption which encompasses both early extinguishing of the outstanding amount by the customer as well as scheduled credit maturity, however given the data spans ten years since loan origination and home loans typically have a contractual maturity upwards of 25 years, the vast majority of

74

Total	80,000	100.0%
Observed Default (event of interest) Observed Redemption (competing risk) Right Censored	1,444 39,608 38,948	1.8% 49.5% 48.7%
Outcome for Loan	Count	Percent

redemptions are early extinguishing of the outstanding amount by the customer. Loans not observed to encounter any event are right-censored; see table 4.3.

Table 4.3: Distribution of Observed and Censored Outcomes

The data contains 17 candidate covariates, covering 4 baseline covariates and 3 timevarying covariates as per table 4.4. Variable 15 (Dynamic Loan to Value Ratio), variable 16 (top-up flag) and variable 17 (Worst Delinquency in Last 6 Months) are the time-varying covariates. Note that including loan-specific time-varying covariates does have implications for out-of-time model application, but for the purpose of testing the methodology they are nevertheless included. We use this dataset to apply our maximum likelihood estimation method to fit competing risks model and compare these results to estimates from the partial likelihood method. Because the partial likelihood method does not estimate the baseline hazard, we calculate this using the Breslow (1972) method. The baseline cause-specific baseline hazards for our maximum likelihood method, as described in equation (4.3), are approximated using piece-wise constant indicator functions for the event of interest and redemption. Knots are used to define the beginning of each of these constant indicator functions, and are selected for each hazard so that an equal number of observed events are contained within each piece-wise constant interval. Note this is only an example and other basis functions could also be applied. Data preparation uses both SAS (2020) and R (2020), with the model estimated in R (2020).

#	Covariate	Description (* = mean corrected)
1	Mortgage Insured	Does the home loan have lenders mortgagee insurance? (Yes / No)?
2	Borrower's Occupation	Professional / Trades / Sales / Other
3	Number of Borrowers	How many borrowers are there for this home loan (either 1 or 2+)?
4	Salary Credits	Do the borrowers deposit salary into the home loan account (Yes/No)?
5	Credit Card	Do the borrowers have a credit card? (Yes/No)
6	Personal Loan	Do the borrowers have a personal loan? (Yes/No)
7	Repayment Method	Are repayments principal and interest (P&I) or interest only (IO)?
8	Repayment Frequency	Are repayments made monthly or fortnightly/other
9	Original Loan Term	What was the original loan term, banded into $(0,5]$, $(5,30]$?
10	Borrower Age	What was the age of the borrower at origination?
11	Product	What is the product, either Investor or Owner Occupier?
12	Borrower Tenure*	How many months has the borrower been a customer $(0,1,2)$?
13	Total Home Loans*	Total number of home loans the borrowers have (integer 1, 2,)
14	Opening Balance*	The original balance when the home loan opened (\$millions)
15	Dynamic Loan to	The current loan balance divided by the current estimate of the value of
	Value Ratio*	the home securing the loan. This is a time-varying covariate, as both the
		numerator and denominator of this ratio vary over time.
16	Top Up Flag	Has the loan been "topped-up" by increasing the original limit. This is a
		time-varying covariate, as all loans begin with Top Up Flag="No", and this
		switches to "Yes" the first time that a loan receives a top-up
17	Worst Delinquency in	The highest number of missed monthly payments within the last 6 months,
	Last 6 Months	lagged by a period of 12 months. This variable can take integer values of
		0, 1, 2,
		(the number of repayments behind a customer is) definitionally begins at
		zero at origination but can change to values greater than zero over time

Table 4.4: Description of the 14 baseline and 3 time-varying covariates in the Home Loan dataset

Tables 4.5 and 4.6 detail estimated regression parameters for the default (the event of interest) and redemption (the competing risk), demonstrating that our maximum likelihood approach produces similar results to the partial likelihood estimation.

Event = Default (customer does not repay monies on time and defaults)							
Covariate	Level	PL Estimate	Std Err	ML Estimate	Std Err		
Mortgage Insured	Yes No	0.1849 0	0.0607	0.1916 0	0.0606		
Borrower Occupation	Professional Trades Other Sales	-0.5551 -0.0388 -0.2332	0.0719 0.0666 0.1013	-0.5546 -0.0384 -0.2309	0.0719 0.0666 0.1013		
Number of Borrowers	1 2+	0 -0 3698	0.0537	0 -0 3711	- 0.0537		
Credit Card	No Yes	$0 \\ -0 2327$	0.0557	$0 \\ -0 2310$	0.0576		
Personal Loan	No Yes	0 3506	$\frac{1}{0}$ 1076	$0 \\ 0 \\ 3525$	0.0270		
Borrowers Tenure (Months) Total Number of Home Loans Opening Balance Worst Delinquency in Last 6 Months Dynamic Loan to Value Ratio	(t-12) (t)	-0.0014 -0.1593 0.4482 2.9079 2.8991	$\begin{array}{c} 0.0003\\ 0.0205\\ 0.1313\\ 0.0732\\ 0.1281\end{array}$	-0.0014 -0.1587 0.4529 2.9117 2.8488	0.0003 0.0205 0.1311 0.0731 0.1254		

Table 4.5: Parameter Estimates for Maximum Likelihood and Partial Likelihood for Default

Event = Redemption (customer repays all monies on time and closes good)							
Covariate	Level	PL Estimate	Std Err	ML Estimate	Std Err		
Number of Borrowers	1	0	-	0	-		
Personal Loan	2+ Yes	0.0595	$0.0108 \\ 0.0247$	0.0595	0.0108 0.0247		
Customer Tenure (Months) Total Home Loans Borrowers Age (Years) Opening Balance	NO	-0.0004 -0.0216 -0.0050 -0.4312	0.0001 0.0032 0.0005 0.0269	-0.0004 -0.0215 -0.0050 -0.4305	0.0001 0.0032 0.0005 0.0269		
Original Loan Term (Years)	(0,5]	0	-	0	-		
Product	[5,50] Investor	-0.2220	0.0133	-0.2212	0.0155		
Top Up Flag (t)	Yes (t) No (t)	-0.1142 0	0.0155	-0.1262 0	- 0.0155 -		

 Table 4.6: Parameter Estimates for Maximum Likelihood and Partial Likelihood for Redemption

To further explore the models fit using maximum liklihood and partial likelihood, the panels in figure 4.3 display estimates for the baseline cause-specific hazard functions, cumulative hazard functions and survival functions for both default and redemption respectively. Each plot compares: (1) the Breslow estimate (grey dashed line); with (2) our maximum likelihood estimate (black solid line) together with the 95% confidence interval calculated using the Hessian matrix (grey shaded area) of our maximum likelihood approach. The plots in figures 4.3a and 4.3b clearly show that our maximum likelihood estimation approach produces less-volatile estimates than the Breslow method. Further, as per figure 4.3a the maximum likelihood approach is able to detect a "humped" baseline cause-specific hazard for the risk of home loan default, as the intensity of default risk begins low before peaking at about 24 months. While the timing and size of this hump could differ between portfolio and jurisdiction, this is a commonly reported feature of credit portfolios (see for example Im et al. (2012) and Bellotti and Crook (2013)). Additionally, as per figure 4.3b the maximum likelihood approach suggests the hazard for redemption remains reasonably flat in comparison the hazard for default, albeit with potentially some minor peaks in intensity. The minor peak at 60 months for the redemption hazard may be driven by the ending of incentive periods, whereby new loans are offered lower interest rates initially but revert to a higher interest rates after an initial few years. What is more striking is that the maximum likelihood approach is better able to detect and measure the increase in variability of the intensity for both loan redemption and default over time, as reflected the 95% confidence intervals starting at their narrowest, increasing as the loans age.

PAPER 2: On Maximum Likelihood Estimation of Competing Risks using the Cause-Specific Semi-Parametric Cox Model with Time-Varying Covariates – an Application to Credit Risk



Figure 4.3: Comparison Between Partial Likelihood and Maximum Likelihood Estimation of Cause-Specific Baseline Hazards and Baseline Cumulative Hazard for Default and Redemption. Shaded Regions are 95% Asymptotic Maximum Likelihood Confidence Intervals.

While the maximum likelihood method produces less volatile estimates of the baseline hazard than using the combination of the partial likelihood and the Breslow estimate, figures 4.3c and 4.3d show the related cumulative hazard functions are very similar, with the Breslow estimate generally falling within the 95% percent confidence interval of the maximum likelihood estimate. The extent of this similarity may be in-part driven by the large sample size of the applied credit risk data (n=80,000), suggesting that numerical implications discussed earlier in section 4.4 of this paper dissipate for much larger sample sizes.

Table 4.7 details calibration and discrimination results for the models estimated using the maximum likelihood method and the partial likelihood method. The first column gives the observed number of defaults while the second and third columns gives respectively the expected number of defaults from each method. The expected number of defaults E[D(T)] up until time *T* needs to recognise the competing risk of redemption. Adapting the approach of Tong et al. (2012), this is achieved using

$$E[D(T)] = \sum_{i=1}^{n} \sum_{t=1}^{T} S_{iD}(t-1) \times S_{iR}(t-1) \times \frac{[S_{iD}(t-1) - S_{iD}(t)]}{S_{iD}(t-1)}$$
(4.14)

where $S_{iD}(t)$ and $S_{iR}(t)$ are the cause-specific survival for subject *i* for default (D) and redemption (R) respectively. The final two columns in table 4.7 display the AUC statistic for both the maximum likelihood and partial likelihood methods, together with the 95% confidences intervals. The AUC values are measured from time zero to the time indicated in the first column of the table. Probability of default can be recovered by dividing the expected number of defaults by the sample size.

Time	Observed	Expected	Expected	AUC (95%CI)	AUC (95%CI)
(months)	Defaults	Defaults ML	Defaults PL	ML	PL+Breslow
12	255	283	266	0.8179 (0.7938, 0.8420)	0.8149 (0.7906, 0.8392)
24	620	702	655	0.7901 (0.7736, 0.8065)	0.7894 (0.7729, 0.8058)
36	885	925	882	0.7847 (0.7702, 0.7992)	0.7846 (0.7700, 0.7991)
48	1,090	1,070	1,029	0.7861 (0.7729, 0.7993) 0.7884 (0.7760, 0.8008)	0.7862 (0.7730, 0.7994)
60	1,244	1,186	1,144		0.7886 (0.7762, 0.8010)
72	1,343	1,256	1,221	0.7871(0.7750, 0.7991)	0.7882 (0.7761, 0.8002)
84	1,388	1,307	1,266	0.7873(0.7754, 0.7991)	0.7874 (0.7755, 0.7992)
96	1,422	1,344	1,304	0.7849(0.7730, 0.7968)	$0.7855 (0.7736, 0.7973) \\ 0.7851 (0.7732, 0.7970) \\ 0.7853 (0.7735, 0.7971)$
108	1,435	1,375	1,332	0.7852(0.7733, 0.7970)	
120	1,440	1,402	1,357	0.7857(0.7739, 0.7974)	

Table 4.7: Calibration and Discrimination Results the Models Estimated using Maximum Likelihood (ML) and Partial Likelihood + Breslow (PL+Bres.) for Default

The conclusion that we draw from the models fit to the credit risk data using maximum likelihood and partial likelihood are that both methods yield highly comparable fitted models. The primary additional insight gained by applying our maximum likelihood method to this data is a clearer and less volatile description of the shape of the underlying hazards for default and redemption together with an estimate of the asymptotic sampling variation of these estimates.

4.5 Conclusion

Credit providers lend money to customers, some of whom may fail to make timely repayments thereby defaulting. An accurate and insightful estimate of the probability of default is a vital input to calculate an institution's minimum regulatory credit capital under the Basel II Accord (BIS, 2006), as well as expected loss under Financial Reporting Standards (IASB, 2014) and (FASB, 2016). Ideally estimating probability of default should recognise the competing risk of redemption as well as time-varying covariates, which requires simultaneous development and estimation of each risk. This paper concerns such a model, detailing a maximum-likelihood approach to estimate the parameters of a competing risks Cox Model with time-varying covariates, and compares this to the prevailing partial likelihood estimation approach for this model in both a simulation study (to understand small-sample behaviour) and in an application to credit risk modelling.

Our maximum likelihood approach provides several computational benefits compared to the partial likelihood estimation, by:

- 1. Jointly estimating both the regression coefficients as well as the baseline hazard using constrained optimisation (ensuring a non-negative baseline hazard);
- Producing asymptotic variance estimates of regression coefficients and the baseline hazard, allowing inferences to be more readily drawn;
- 3. Producing a baseline hazard estimate with far less volatility than the Breslow estimator (which uses the results of the partial likelihood estimation as input).

In a simulation study, these benefits are explored and show our estimation approach can accurately recover the true parameter estimate in small samples with heavy censoring.

In an application to credit risk, these benefits are explored by comparing various aspects of a model fit using maximum likelihood estimation to a model fit using partial likelihood estimation, with each designed to predict the time to loan default using time-varying covariates and in the presence of the competing risk of redemption.

The results show our method provides a comparable model fit to that under the partial likelihood approach, but our maximum likelihood approach provides a key benefit of enhancing clarity of the shape of the baseline hazards for default and redemption. While the partial likelihood approach (in combination with the Breslow estimator) produces a baseline hazard estimate with a high degree of volatility, the maximum likelihood approach returns a less volatile estimate together with an associated standard error which we can use to construct and display a confidence intervals. For the data at hand, this allows us to locate evidence of the "humped" hazard along with its significance, a feature that has been previously reported

in other credit risk data (see for example Im et al. (2012) and Bellotti and Crook (2013)). Similarly, our method allows a clear picture to be formed on the hazard or redemption together with its significance, which for this data is comparably flatter than the hazard for default but with a minor peak at approximately 60 months. Our approach also quantifies the increase in variability of the intensity for both loan redemption and default over time. On this basis, we feel our maximum likelihood estimation approach is a plausible alternative for credit-granting institutions to apply to estimate probability of default.

Further directions we would like to take this research includes using macroeconomic time-varying covariates to seek to improve the model fit, implementation of covariate splines and release of our source code in a published R package.

Our paper is supported by a supplementary materials document that thoroughly details the derivation of the gradient and Hessian necessary for our maximum likelihood approach, outlines the computational scheme to estimate the model parameters and proves the asymptotic properties of our estimates.

References

- Austin, P. (2012). Generating survival times to simulate cox proportional hazards models with time-varying covariates. *Statistics in Medicine*, 31(29), 3946-3958. https://doi.org/10.1002/sim.5452
- Austin, P., Lee, D., Fine, J. (2002). Introduction to the Analysis of Survival Data in the Presence of Competing Risks. *Circulation*, 133(6), 601-609. https://doi.org/10.1161/CIRCULATIONAHA.115.017719
- Banasik, J., Crook, J., Thomas, L. (1999). Not if but when will borrowers default. *Journal* of the Operational Research Society, 50(12), 1185-1190. https://doi.org/10.2307/3010627
- Bellotti, T. and Crook, J. (2009). Credit scoring with macroeconomic variables using survival analysis. *Journal of the Operational Research Society*, 60(12), 1699-1707. https://doi.org/10.1057/jors.2008.130
- Bellotti, T., and Crook, J. (2013). Forecasting and stress testing credit card default using dynamic models. International Journal of Forecasting, 29(4), 563-574. https://doi.org/10.1016/j.ijforecast.2013.04.003
- Bender, R., Augustin, T., and Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. *Statistics in Medicine*, 24(11), 1713-1723. https://doi.org/10.1002/sim.2059
- Beyersmann, J., Latouche, A., Buchholz, A., Schumacher, M. (2009). Simulating competing risks data in survival analysis *Statistics in Medicine*, 28(6), 956-971. https://doi.org/10.1002/sim.3516

- BIS (2006). International Convergence of Capital Measurement and Capital Standards A Revised Framework Comprehensive Version *Bank for International Settlements*. Basel Framework
- Breslow, N. E. (1972). Contribution to the discussion of paper by D.R. Cox. *Journal of the Royal Statistical Society: Series B*, 34, 216–217 https://www.jstor.org/stable/2985181
- Chan, R. H. and Ma, J. (2012). A multiplicative iterative algorithm for box-constrained penalized likelihood image restoration. *IEEE Trans. Image Processing*, 21, 3168-3181. https://doi.org/10.1109/TIP.2012.2188811
- Cox, D. (1972). Regression models and life tables. *Journal of the Royal Statistical Society, Series B*, 34(2), 187-220. https://doi.org/10.1111/j.2517-6161.1972.tb00899.x
- Cox, D. (1975). Partial likelihood. *Biometrika*, 62(2), 269-276. https://doi.org/10.1093/biomet/62.2.269
- Crowley, J. and Hu, M. (1977). Covariance analysis of heart transplant survival data. *Journal of the American Statistical Association*, 72(357), 27-36. https://doi.org/10.1080/01621459.1977.10479903
- Deng, Y., Quigley, J., Van Order, R. (2000) Mortgage termination heterogeneity and the exercise of mortgage options. Econometrica, 68(2), 275-307. https://doi.org/10.1111/1468-0262.00110
- Financial Accounting Standards Board (2016). Financial Instruments—Credit Losses (Topic 326) No. 2016-13 Measurement of Credit Losses on Financial Instruments. FASB Accounting Standards
- Fine, P. and Gray, R. (1999). A Proportional Hazards Model for the Subdistribution of a Competing Risk. *Journal of the American Statistical Association*, 94(466), 496-509. https://doi.org/10.1080/01621459.1999.10474144
- Hosmer, D., Lemeshow, S., and May, S. (2008). Applied Survival Analysis: Regression Modeling of Time to Event Data, (2nd ed.). Wiley-Interscience, Hoboken, New Jersey.
- IASB. (2014). International Financial Reporting Standard 9. https://www.ifrs.org/issuedstandards/list-of-standards/ifrs-9-financial-instruments/

- Im, J., Apley, D., Qi, C., Shan, X. (2012) A time-dependent proportional hazards survival model for credit risk analysis. *Journal of the Operational Research Society*, 63(3), 306-321. https://doi.org/10.1057/jors.2011.34
- Kaplan. E. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *The Journal of the American Statistical Association*, 53(282), 457-481. https://doi.org/10.1080/01621459.1958.10501452
- Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. [Master's thesis, M.Sc. Dissertation. Dept. of Mathematics], Univ. of Chicago.
- Kuhn, H. and Tucker (1951). Nonlinear programming. *Proceedings of 2nd Berkeley Symposium. Berkeley*, pp481-492.
- Luenberger, D. and Ye, Y. (2008). *Linear and Nonlinear Programming, 3 ed.* Springer, New York, NY.
- Ma, J. (2010). Positively constrained multiplicative iterative algorithm for maximum penalized likelihood tomographic reconstruction. *IEEE Transactions on Nuclear Science*, 57(1), 181-192. https://doi.org/10.1109/TNS.2009.2034462
- Ma, J., Heritier, S., and Lô, S. (2014). On the maximum penalized likelihood approach for proportional hazard models with right censored survival data. *Computational Statistics and Data Analysis*, 74, 142-156. https://doi.org/10.1016/j.csda.2014.01.005
- Pintilie, M. (2006). Competing Risks a Practical Perspective John Wiley and Sons, West Sussex, England.
- Prentice, R. L., Kalbfleisch, J. D., Peterson, A. V., Flournoy, N., Farewell, V. T. and Breslow, N. E. (1978). The analysis of fa7ilure times in the presence of competing risks. *Biometrics*, 34(4), 541-554. https://doi.org/10.2307/2530374
- Putter, H., Fiocco, M. and Geskus, R. B. (2007). Tutorial in biostatistics: competing risks and multi-state models. *Statistics in Medicine*, 26(11), 2389-2430. https://doi.org/10.1002/sim.2712
- R Core Team. (2020). *R* 4.0.0: A Language and Environment for Statistical Computing. *R* Foundation for Statistical Computing. http://www.R-project.org

- Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science*, 3(4), 425-441. https://doi.org/10.1214/ss/1177012761
- SAS Institute Inc. (2020). SAS Software, Version 9.4. http://www.sas.com
- Stepanova, M. and Thomas, L. (2002). Survival analysis methods for personal loan data. *Operations Research*, 50(2), 277-289. https://doi.org/10.1287/opre.50.2.277.426
- Thackham, M. and Ma, J. (2020). On maximum likelihood estimation of the semi-parametric Cox model with time-varying covariates. *Journal of Applied Statistics*, 47(9), 1511-1528. https://doi.org/10.1080/02664763.2019.1681946
- Therneau, T. M. (2019). *A Package for Survival Analysis in R*. (R package version 3.1-6). https://CRAN.R-project.org/package=survival
- Therneau, T., Crowson, C., and Atkinson, E. (2015). Using time dependent covariates and time dependent coefficients in the Cox model. R CRAN Vignette. https://cran.rproject.org/web/packages/survival/vignettes/timedep.pdf
- Tong, T., Mues, C., and Thomas, L. (2012). Mixture cure models in credit scoring: If and when borrowers default. *European Journal of Operational Research*, 218(1). 132-139. https://doi.org/10.1016/j.ejor.2011.10.007
- Wong, W. H. and Severini, T. A. (1991). On maximum likelihood estimation in infinite dimensional parameter spaces. Ann. Statist., 19(2), 603-632. https://doi.org/10.1214/aos/1176348113
- Wycinka, E. (2019) Competing Risk Models of Default in the Presence of Early Repayments. Econometrica, 23(2), 99–120. https://doi.org/10.15611/eada.2019.2.07
- Xu, J., Ma, J., Connors, M. H. and Brodaty, H. (2018). Proportional hazard model estimation under dependent censoring using copulas and penalized likelihood. *Statistics in Medicine*, 37(14), 2238-2251. https://doi.org/10.1002/sim.7651
- Yu, Y. and Ruppert, D. (2002). Penalized spline estimation for partially linear single index models. J. Amer. Statist. Assoc., 97(460), 1042-1054. https://doi.org/10.1198/016214502388618861

Zhang Z., Reinikainen J., Adeleke K.A., Pieterse M.E., Groothuis-Oudshoorn C.G. (2018). *Time-varying covariates and coefficients in Cox regression models*. *Annals of Translational Medicine*, 6(7), 121. https://doi.org/10.21037/atm.2018.02.12

4.6 Supplementary Material

(A) Hessian matrix

The second derivatives with respect to β_g and γ_g are:

$$\frac{\partial^2 l}{\partial \boldsymbol{\beta}_g \partial \boldsymbol{\beta}_g^T} = -\mathbf{X}^T \mathbf{A}_g \mathbf{X}, \quad \frac{\partial^2 l}{\partial \boldsymbol{\gamma}_g \partial \boldsymbol{\gamma}_g^T} = -\mathbf{Z}^T \mathbf{B}_g \mathbf{Z}.$$

The other second derivatives are:

$$\frac{\partial^2 l}{\partial \theta_{gu} \partial \theta_{gv}} = -\sum_{i=1}^n \delta_{ig} \frac{\psi_{gu}(y_i)\psi_v(y_i)}{h_{0g}(y_i)^2}, \quad \frac{\partial^2 l}{\partial \beta_{gj} \partial \theta_{gu}} = -\sum_{i=1}^n \Psi_{gui}^*(y_i)e^{\mathbf{x}_i^T \beta_g} x_{ij},$$
$$\frac{\partial^2 l}{\partial \gamma_{gb} \partial \theta_{gu}} = -\sum_{i=1}^n \Psi_{guib}^{**}(y_i)e^{\mathbf{x}_i^T \beta_g} x_{ij}, \quad \frac{\partial^2 l}{\partial \beta_{gj} \partial \gamma_{gb}} = -\sum_{i=1}^n H_{0gib}^{**}(y_i)e^{\mathbf{x}_i^T \beta_g} x_{ij},$$

where, if $t \in (t_{id}, t_{i,d+1}]$,

$$\Psi_{guib}^{**}(t) = \sum_{a=1}^{d} [\Psi_{gu}(t_{i,a+1}) - \Psi_{gu}(t_{i,a})] e^{\mathbf{z}_{ia}^{T} \boldsymbol{\gamma}_{g}} z_{iab}$$
$$H_{0gib}^{**}(t) = \sum_{a=1}^{d} [H_{0g}(t_{i,a+1}) - H_{0g}(r_{i,a})] e^{\mathbf{z}_{ia}^{T} \boldsymbol{\gamma}_{g}} z_{iab}$$

Constrained Optimisation Algorithm

The Karush–Kuhn–Tucker (KKT) (Karush (1939), Kuhn and Tucker (1951)) first-order necessary conditions for the constrained optimal solution of β_g , γ_g and $\theta_g \ge 0$ are

$$\frac{\partial l}{\partial \beta_{gj}} = 0, \tag{4.15}$$

$$\frac{\partial l}{\partial \gamma_{gb}} = 0, \tag{4.16}$$

$$\frac{\partial l}{\partial \theta_{gu}} = 0 \text{ if } \theta_{gu} > 0 \text{ or } \frac{\partial l}{\partial \theta_{gu}} < 0 \text{ if } \theta_{gu} = 0 \tag{4.17}$$

where g = 1, ..., G represent the G competing risks, j = 1, ..., p, b = 1, ..., q and $u = 1, ..., m_g$.

For time-varying covariates, special care must be taken for data format and data entry. We adopt the "long-format" data frame to accommodate time-varying covariates in our R program, with Table 4.8 showing a simple example to elaborate this data frame. It details the data frame structure for the first two subjects who are exposed to only two risks, however the structure generalises for more than two risks. Critically, this "long-format" creates a record for each subject when any time-varying covariate subject *i* is measured. Additionally, it contains one status column for each risk (columns for δ_{ig}) the subjects are exposed to, so for this example there are two status columns denoted status₁ and status₂. Focussing on Subject 1, it has time-varying covariates that are measured twice at times t_{12} and t_{13} , and experiences an event for risk 1 at time y_1 . As subject 1 experiences an event for risk 1, the column status₁ is set to 0 at all time points. Turning focus to subject 2, it has time-varying covariates measured at time t_{22} and then experiences an event for risk 2 at time at y_2 . As subject 2 experiences an event for risk 2, the column status₂ is set to 1 for the final record and 0 for all previous time points.

subject	start	end	$status_1$	$status_2$	$z_1(t)$		$z_q(t)$
1	$t_{11} t_{12} t_{12}$	$t_{12} t_{13}$	0 0 1	0 0 0	Z111 Z121	 	z_{11q} z_{12q}
2	t_{13} t_{21} t_{22}	$\begin{array}{c} y_1 \\ t_{22} \\ y_2 \end{array}$	0 0	0 1	z131 z211 z221	 	z13q Z21q Z22q
3							
÷	÷	÷	÷	÷	÷	÷	÷

Table 4.8: Example of time-varying covariate data frame

The long-format data frame can be easily created in R (2020). For example, Zhang et al. (2018) demonstrate construction of such as data frame with the assistance of the survival::tmerge() function from the survival package (Therneau et al., 2015).

Many available algorithms for constrained optimisation (for example Luenberger and Ye (2008)) are less efficient for our problem when some m_g are large, such as if one uses indicator basis functions where each bin contains just one event time. We therefore develop an algorithm which is efficient for a large m_g . We adopt the following alternating iterative strategy to solve equations given in (4.15), (4.16) and (4.17), and update β_g , γ_g and θ_g for each risk g in turn. With a given g, beginning with the current estimates $\beta_g^{(k)}$, $\gamma_g^{(k)}$ and $\theta_g^{(k)}$ at iteration k, iteration k + 1 comprises the following alternating steps:

1: compute $\beta_g^{(k+1)}$ so that $l(\beta_g^{(k+1)}, \gamma_g^{(k)}, \theta_g^{(k)}) \ge l(\beta_g^{(k)}, \gamma_g^{(k)}, \theta_g^{(k)})$; 2: compute $\gamma_g^{(k+1)}$ so that $l(\beta_g^{(k+1)}, \gamma_g^{(k+1)}, \theta_g^{(k)}) \ge l(\beta_g^{(k+1)}, \gamma_g^{(k)}, \theta_g^{(k)})$; 3: compute $\theta_g^{(k+1)} \ge 0$ so that $l(\beta_g^{(k+1)}, \gamma_g^{(k+1)}, \theta_g^{(k+1)}) \ge l(\beta_g^{(k)}, \gamma_g^{(k)}, \theta_g^{(k)})$. These conditions assure $l(\beta_g^{(k+1)}, \gamma_g^{(k+1)}, \theta_g^{(k+1)}) \ge l(\beta_g^{(k)}, \gamma_g^{(k)}, \theta_g^{(k)})$ at the end of iteration k+1, which is a key requirement for convergence of this algorithm. Steps 1 and 2 are solved by the Newton algorithm incorporating line search steps, and Step 3 is solved by a multiplicative-iterative (MI) algorithm (e.g. Chan and Ma (2012) and Ma (2010)) designed to respect the non-negative constraints on θ_g . We call this algorithm the Newton-MI algorithm similar to Ma et al. (2014).

To update $\boldsymbol{\beta}_g$, we employ one iteration of the Newton algorithm with line search. Starting with $\boldsymbol{\beta}_g^{(k)}$ and using a line-search with step size $\omega_{1g}^{(k)} \in (0, 1]$, we have

$$\boldsymbol{\beta}_{g}^{(k+1)} = \boldsymbol{\beta}_{g}^{(k)} + \omega_{1g}^{(k)} (\mathbf{X}^{T} \mathbf{A}_{g}^{(k)} \mathbf{X})^{-1} \mathbf{X}^{T} (-\mathbf{A}_{g}^{(k)} \mathbf{1}_{n} + \boldsymbol{\delta}_{g}), \qquad (4.18)$$

where **X** (of size $n \times p$) is the model matrix of time fixed covariates and its *i*-th row is given by \mathbf{x}_i^T , \mathbf{A}_g is a diagonal matrix given by $\mathbf{A}_g = \text{diag}(H_{0g1}^*(y_1)e^{\mathbf{x}_1^T\boldsymbol{\beta}_g}, \ldots, H_{0gn}^*(y_n)e^{\mathbf{x}_n^T\boldsymbol{\beta}_g})$, $\mathbf{1}_n$ is an *n*-vector of 1's and $\boldsymbol{\delta}_g$ is an *n*-vector for $\boldsymbol{\delta}_{ig}$'s for the competing risk *g*. Matrix $\mathbf{A}_g^{(k)}$ is \mathbf{A}_g with $\boldsymbol{\beta}_g = \boldsymbol{\beta}_g^{(k)}$, $\boldsymbol{\gamma}_g = \boldsymbol{\gamma}_g^{(k)}$ and $\boldsymbol{\theta}_g = \boldsymbol{\theta}_g^{(k)}$. Matrix $\mathbf{X}^T \mathbf{A}_g \mathbf{X}$ is the negative Hessian of $l(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\beta}_g$. Let $l_g(\boldsymbol{\beta}_g, \boldsymbol{\gamma}_g, \boldsymbol{\theta}_g)$ be the part of log-likelihood (as per equation (9) of the main paper) that is related to risk *g* only, and clearly we have

$$l(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta}) = \sum_{g=1}^{G} l_g(\boldsymbol{\beta}_g, \boldsymbol{\gamma}_g, \boldsymbol{\theta}_g)$$

since parameters of different risks are fully separated. The line search parameter $\omega_{1g}^{(k)}$ helps to achieve $l_g(\boldsymbol{\beta}_g^{(k+1)}, \boldsymbol{\gamma}_g^{(k)}, \boldsymbol{\theta}_g^{(k)}) \ge l_g(\boldsymbol{\beta}_g^{(k)}, \boldsymbol{\gamma}_g^{(k)}, \boldsymbol{\theta}_g^{(k)})$.

Similarly, the Newton method with line search is also applied to update γ_g . First, let $\mathbf{Z} = [\mathbf{Z}_1^T, \dots, \mathbf{Z}_n^T]^T$, where $\mathbf{Z}_i = [\mathbf{z}_{i1}, \dots, \mathbf{z}_{i,n_i+1}]^T$. Here, each \mathbf{z}_{ia} is a vector defined with equation (7) of the main paper. Matrix \mathbf{Z} is in fact the model matrix associated with the time-varying covariates, and it has the dimension of $N \times q$, where $N = \sum_i n_i$. Let $\mathbf{B}_g = \text{diag}(e^{\mathbf{x}_1^T \boldsymbol{\beta}_g} \mathbf{B}_{g1}, \dots, e^{\mathbf{x}_n^T \boldsymbol{\beta}_g} \mathbf{B}_{gn})$, where \mathbf{B}_{gi} is a diagonal matrix of size $n_i \times n_i$ with diagonal elements $[H_{g0}(t_{i,a+1}) - H_{g0}(t_{i,a})]e^{\mathbf{z}_{ia}^T \boldsymbol{\gamma}_g}$ for $a = 1, \dots, n_i$. Let $\boldsymbol{\zeta}_g$ be an *N*-vector for

status_p column; see Table 4.8. Namely, $\zeta_g = [\zeta_{g11}, \ldots, \zeta_{g1n_1}, \ldots, \zeta_{gn1}, \ldots, \zeta_{gnn_n}]^T$, where $\zeta_{gia} = 1$ only if $a = n_{gi}$ and $\delta_{ig} = 1$; otherwise, $\zeta_{gia} = 0$. We again conduct one iteration of the Newton algorithm with line search to update γ_g :

$$\boldsymbol{\gamma}_{g}^{(k+1)} = \boldsymbol{\gamma}_{g}^{(k)} + \omega_{2g}^{(k)} (\mathbf{Z}^{T} \mathbf{B}_{g}^{(k)} \mathbf{Z})^{-1} \mathbf{Z}^{T} (-\mathbf{B}_{g}^{(k)} \mathbf{1}_{N} + \boldsymbol{\zeta}_{g}),$$
(4.19)

where $\omega_{2g}^{(k)} \in (0,1]$ is a line search step size and $\mathbf{B}_g^{(k)}$ denotes \mathbf{B}_g but with $\boldsymbol{\beta}_g = \boldsymbol{\beta}_g^{(k+1)}$, $\boldsymbol{\gamma}_g = \boldsymbol{\gamma}_g^{(k)}$ and $\boldsymbol{\theta}_g = \boldsymbol{\theta}_g^{(k)}$. Note that matrix $\mathbf{Z}^T \mathbf{B}_g \mathbf{Z}$ is the negative Hessian of $l(\boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\gamma}_g$. Note that $\omega_{2g}^{(k)}$ is used to achieve $l_g(\boldsymbol{\beta}_g^{(k+1)}, \boldsymbol{\gamma}_g^{(k+1)}, \boldsymbol{\theta}_g^{(k)}) \ge l_g(\boldsymbol{\beta}_g^{(k+1)}, \boldsymbol{\gamma}_g^{(k)}, \boldsymbol{\theta}_g^{(k)})$.

Finally for updating θ_g , we adopt the multiplicative-iterative (MI) algorithm (see, for example, Chan and Ma (2012), Ma et al. (2014) and Thackham and Ma (2020)) which respects the non-negative constraint on θ_g and also is very easy to implement. Towards this, we let both \mathbf{C}_g and \mathbf{C}_g^* be $n \times m_g$ matrices; and their (i, u)-th elements are $\psi_{gu}(y_i)$ and $\Psi_{gui}^*(y_i)$ respectively. Let \mathbf{f}_g be an *n*-vector for $e^{\mathbf{x}_i^T \boldsymbol{\beta}_g}$. The MI algorithm updates θ_g according to

$$\boldsymbol{\theta}_{g}^{(k+1)} = \boldsymbol{\theta}_{g}^{(k)} + \omega_{3g}^{(k)} \mathbf{S}_{g}^{(k)} \left(\mathbf{C}_{g}^{T} [\mathbf{D}_{g}^{(k)}]^{-1} \boldsymbol{\delta}_{g} - [\mathbf{C}_{g}^{*(k)}]^{T} \mathbf{f}_{g}^{(k)} \right),$$
(4.20)

where \mathbf{D}_g and \mathbf{S}_g are diagonal matrices with elements $h_{0g}(y_i)$ and $\theta_{gu}/(\sum_i \Psi_{gui}^*(y_i)e^{\mathbf{x}_i^T \boldsymbol{\beta}_g} + \varepsilon_g)$ respectively. In (4.20), ε_g is a small threshold for avoiding the corresponding denominator being zero. The step size $\omega_{3g}^{(k)} \in (0,1]$ is used to achieve $l_g(\boldsymbol{\beta}_g^{(k+1)}, \boldsymbol{\gamma}_g^{(k+1)}, \boldsymbol{\theta}_g^{(k+1)}) \geq l_g(\boldsymbol{\beta}_g^{(k+1)}, \boldsymbol{\gamma}_g^{(k+1)}, \boldsymbol{\theta}_g^{(k)})$.

The line search steps can be efficiently conducted using, for example, the Armijo rule (Luenberger and Ye, 2008). This algorithm is very easy to implement. The updating formula for γ_g naturally suggests the "long" data format, similar to the R "survival" package (Therneau, 2019) for time dependent covariates as demonstrated in Table 4.8. The "start" and "end" times are needed to select the \mathbf{B}_g matrix and compute Ψ^*_{gui} values and the ζ_g vector is determined solely from the corresponding "status" column for risk g.

Convergence of this algorithm can be established following Chan and Ma (2012). In particular, we can prove that when $\mathbf{A}_g^{1/2}\mathbf{X}$ and $\mathbf{B}_g^{1/2}\mathbf{Z}$ have full column rank, then (i) if $\boldsymbol{\theta}_g^{(k)}$ is non-negative then $\boldsymbol{\theta}_g^{(k+1)}$ is also non-negative, and (ii) under certain regularity conditions, this algorithm converges to a solution satisfying the KKT conditions (4.15) – (4.17). Asymptotic results of our method are supplied in Appendix 4.6.

Asymptotic Properties

In this section we discuss some asymptotic results for the maximum likelihood estimates. These results are useful for performing inferences without relying on computational demanding methods such as bootstrapping.

For risk g, let $(\beta_{0g}, \gamma_{0g}, h_{00g}(t))$ denote the true parameters associated with this risk. Recall we $h_{ng}(t) = \sum_{u=1}^{m_g} \theta_{gu} \psi_{gu}(t)$ to approximate the baseline hazard $h_{0g}(t)$ using m_g basis functions. Let $\hat{\beta}_g, \hat{\gamma}_g$ and $\hat{\theta}_g$ be the ML estimates. Define $\hat{h}_{ng}(t) = \sum_{u=1}^{m} \hat{\theta}_{gu} \psi_{gu}(t)$.

Following Xu et al. (2018), strong consistency results for $\hat{\beta}_g$, $\hat{\gamma}_g$ and $\hat{h}_{ng}(t)$ can be established when $m_g \to \infty$ and $n \to \infty$ but $m_g/n \to 0$, with results given in Theorem 3 below. Theorem 4 provides a more useful asymptotic normality result for $\hat{\beta}_g$, $\hat{\gamma}_g$ and $\hat{\theta}_g$ for fixed m_g . In practice, one can only has a large but finite *n*, and thus a fixed m_g is a reasonable assumption. However, the size of m_g can vary with *n*. The simulation results in Section 3 of the main paper demonstrate that the asymptotic variance in Theorem 4 is accurate when compared with the variance obtained from the Monte Carlo method.

The proofs for Theorems 3 and 4 are omitted as they are very similar to the corresponding proofs in Xu et al. (2018). The results in these theorems require some regularity conditions similar to these in Xu et al. (2018). We first state assumptions needed for Theorem 3.

- A1. Matrices **X** and **Z** are bounded and both $E(\mathbf{X}\mathbf{X}^T)$ and $E(\mathbf{Z}\mathbf{Z}^T)$ are non-singular.
- A2. For the function $h_{0g}(t)$, its corresponding coefficient vector θ_g is in a compact subset of R^{m_g} .
- A3. Assume that for any baseline hazard $h_{0g}(t)$, there exists a $h_{ng}(t)$ such that $\max_t |h_{ng}(t) h_{g0}(t)| \to 0$ as $m_g \to \infty$ and $n \to \infty$ but $m_g/n \to 0$.

Note that assumption A3 can be satisfied according to Proposition 2.8 of DeBoor and Daniel (1974).

Theorem 3 Assume Assumptions A1 - A3 hold. Assume there exists an interval [a, b], where $0 \le a < b < \infty$, such that $h_{0g}(t)$ is bounded and has up to $r \ge 1$ derivatives over [a, b]. Assume the number of basis functions satisfy $m_g = n^{\upsilon}$, where $0 < \upsilon < 1$. Then, when $n \to \infty$,

- 1. $\|\widehat{\boldsymbol{\beta}}_g \boldsymbol{\beta}_{0g}\| \to 0$ and $\|\widehat{\boldsymbol{\gamma}}_g \boldsymbol{\gamma}_{0g}\| \to 0$ almost surely, and
- 2. $\sup_{t \in [a,b]} |\widehat{h}_{ng}(t) h_{00g}(t)| \to 0$ almost surely.

We next develop a asymptotic normality result for $\hat{\beta}_g$, $\hat{\gamma}_g$ and $\hat{\theta}_g$ assuming a fixed m_g , and therefore m_g is retained in the asymptotic results. This result is capable to provide inferences on the baseline hazard and the rational for a fixed m_g is similar to the argument in Yu and Ruppert (2002). The maximum likelihood estimates $\hat{\beta}_g$, $\hat{\gamma}_g$ and $\hat{\theta}_g$ can still achieve \sqrt{n} convergence rate in this context. The fix m_g assumption lies somewhere between parametric and non-parametric modelling as there are freedoms in selecting a value for m_g . A major benefit is that this asymptotic normality can be used to make inferences involving β_g , γ_g and $h_{g0}(t)$. This will be helpful in, for example, inferences on survival probabilities or predictions.

Let $\eta_g = (\theta^T, \beta_g^T, \gamma_g^T)^T$, whose length is $m_g + p + q$, and it makes the log-likelihood a function of $\eta = (\eta_1^T, \dots, \eta_G^T)^T$. The maximum likelihood estimate of all η_g , denoted by $\hat{\eta}_g$, are obtained by maximizing $l(\eta)$ with the constraint $\theta_g \ge 0$ for all g. In developing this asymptotic normality, we must allow the possibility of active constraints (i.e. some $\theta_{gu} = 0$); otherwise, a non-positive definite information matrix can be obtained if some θ_{gu} estimates are 0.

The way to allow for active constraints in the asymptotic results has been developed, for example, in Xu et al. (2018) and Thackham and Ma (2020). Suppose for risk g, there are c_g active $\theta_g \ge 0$ constraints in the maximum likelihood estimates. We define a matrix \mathbf{U}_g in Assumption B5 below to indicate the active constraints, which satisfies $\mathbf{U}_g^T \mathbf{U}_g =$ $\mathbf{I}_{(m_g+p+q-c_g)\times(m_g+p+q-c_g)}$.

The asymptotic normality results for the maximum likelihood estimate $\hat{\eta}_g$ require the following regularity assumptions.

Assumptions:

- B1. For $t \in [a, b]$, random vectors $(y_i, \delta_{i1}, \ldots, \delta_{iG}, \mathbf{x}_i^T, \{\mathbf{z}_i^T(\tau), \tau \leq t\}), i = 1, \ldots, n$, are independent and identically distributed.
- B2. The space Ξ_g for η_g is compact for all g.
- B3. Let $\eta_0 = (\eta_{01}^T, \dots, \eta_{0G}^T)^T$ be the true parameter vector. Then, $\lim_{n \to \infty} n^{-1} l(\eta)$ exists and has a unique maximum at η_0 .
- B4. $l(\boldsymbol{\eta})$ is bounded and is twice continuously differentiable in a neighbourhood of $\boldsymbol{\eta}_0$, and the matrix $\mathbf{F}(\boldsymbol{\eta}) = \lim_{n \to \infty} n^{-1} \partial l^2 / \partial \boldsymbol{\eta} \partial \boldsymbol{\eta}^T$ exist. Since parameters for different risks are fully separated in the log-likelihood function l, \mathbf{F} is a block diagonal matrix with blocks $\mathbf{F}_g(\boldsymbol{\eta}_g) = \lim_{n \to \infty} n^{-1} \partial l_g^2 / \partial \boldsymbol{\eta}_g \partial \boldsymbol{\eta}_g^T$.

B5. Assume there are c_g active constraints from θ_g . Let \mathbf{U}_g be a matrix whose rows take zero values corresponding to the active constraints and other rows form an identity matrix. \mathbf{U}_g has the dimension $(m_g + p + q) \times (m_g + p + q - c_g)$. Assume the matrix $\mathbf{U}_g^T \mathbf{F}_g(\boldsymbol{\eta}_g) \mathbf{U}_g$ is invertible in a neighbourhood of $\boldsymbol{\eta}_{0g}$.

Theorem 4 Assume Assumptions B1 - B5 hold. For risk g assume there are c_g active constraints in the maximum likelihood estimate of θ_g and the corresponding \mathbf{U}_g is defined as in Assumption B5. Then, when $n \to \infty$,

- 1. The constrained MPL estimate $\hat{\eta}_{g}$ is consistent for η_{0g} , and
- 2. $\sqrt{n}(\widehat{\boldsymbol{\eta}}_g \boldsymbol{\eta}_{0g})$ converges in distribution to $N_g(\mathbf{0}, \widetilde{\mathbf{F}}_g(\boldsymbol{\eta}_{g0})^{-1})$, where the covariance matrix given by is $\widetilde{\mathbf{F}}_g(\boldsymbol{\eta}_g)^{-1} = \mathbf{U}_{\mathbf{g}}(\mathbf{U}_{\mathbf{g}}^T \mathbf{F}_{\mathbf{g}}(\boldsymbol{\eta}_g) \mathbf{U}_{\mathbf{g}})^{-1} \mathbf{U}_{\mathbf{g}}^T$.

Matrix $\widetilde{\mathbf{F}_g}(\boldsymbol{\eta}_g)^{-1}$ is in fact very easy to compute. Firstly, $\mathbf{U_g}^T \mathbf{F}_g(\boldsymbol{\eta}_g) \mathbf{U}_g$ is obtained simply by deleting the rows and columns of $\mathbf{F}_g(\boldsymbol{\eta}_g)$ associated with the active constraints. Then $(\mathbf{U}_g^T \mathbf{F}_g(\boldsymbol{\eta}_g) \mathbf{U}_g)^{-1}$ is calculated. Finally, $\widetilde{\mathbf{F}_g}(\boldsymbol{\eta}_g)^{-1}$ is obtained by padding $(\mathbf{U}_g^T \mathbf{F}_g(\boldsymbol{\eta}_g) \mathbf{U}_g)^{-1}$ with zeros at the positions of the deleted rows and columns. Implementation of the results in Theorem 4 demands that the active constraints must be identified. We adopt the following strategy. If the maximum likelihood estimates of $\hat{\theta}_{gu}$ is either exactly zero or very close to zero with gradient of less than (say) -10^{-2} then we treat it as active.

The simulation results reported in Section 3 of the main paper demonstrate that biases in the maximum likelihood estimates are usually negligible and the asymptotic variances of the estimates are accurate.

References

- Chan, R. H. and Ma, J. (2012). A multiplicative iterative algorithm for box-constrained penalized likelihood image restoration. *IEEE Trans. Image Processing*, 21(7), 3168-3181. https://ieeexplore.ieee.org/document/6156784
- Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. [Master's thesis, M.Sc. Dissertation. Dept. of Mathematics], Univ. of Chicago.
- Kuhn, H. and Tucker (1951). Nonlinear programming. *Proceedings of 2nd Berkeley Symposium. Berkeley*, pp481-492.
- Luenberger, D. and Ye, Y. (2008). *Linear and Nonlinear Programming, 3 ed.* Springer, New York, NY.
- Ma, J. (2010). Positively constrained multiplicative iterative algorithm for maximum penalized likelihood tomographic reconstruction. *IEEE Transactions on Nuclear Science*, 57(1), 181-192. https://doi.org/10.1109/TNS.2009.2034462
- Ma, J., Heritier, S., and Lô, S. (2014). On the maximum penalized likelihood approach for proportional hazard models with right censored survival data. *Computational Statistics and Data Analysis*, 74, 142-156. https://doi.org/10.1016/j.csda.2014.01.005
- R Core Team. (2020). *R* 4.0.0: A Language and Environment for Statistical Computing. *R* Foundation for Statistical Computing. http://www.R-project.org
- Thackham, M. and Ma, J. (2020). On maximum likelihood estimation of the semi-parametric Cox model with time-varying covariates. *Journal of Applied Statistics*, 47(9), 1511-1528. https://doi.org/10.1080/02664763.2019.1681946
- Therneau, T. M. (2019). *A Package for Survival Analysis in R*. (R package version 3.1-6). https://CRAN.R-project.org/package=survival
- Therneau, T., Crowson, C., and Atkinson, E. (2015). Using time dependent covariates and time dependent coefficients in the Cox model. R CRAN Vignette. https://cran.rproject.org/web/packages/survival/vignettes/timedep.pdf
- Xu, J., Ma, J., Connors, M. H. and Brodaty, H. (2018). Proportional hazard model estimation under dependent censoring using copulas and penalized likelihood. *Statistics in Medicine*, 37, 2238-2251. https://doi.org/10.1002/sim.7651
- Yu, Y. and Ruppert, D. (2002). Penalized spline estimation for partially linear single index models. J. Amer. Statist. Assoc., 97(460), 1042-1054. https://doi.org/10.1198/016214502388618861
- Zhang Z., Reinikainen J., Adeleke K.A., Pieterse M.E., Groothuis-Oudshoorn C.G. (2018). *Time-varying covariates and coefficients in Cox regression models*. *Annals of Translational Medicine*, 6(7), 121. https://doi.org/10.21037/atm.2018.02.12

PAPER 3: Maximum Likelihood Estimation of the Mixture Cure Semi-Parametric Cox Model – an Application to Credit Risk

5

This chapter is a manuscript submitted in consideration for publication in October 2020.

Thackham, M. and Ma, J. Maximum Likelihood Estimation of the Mixture Cure Semi-Parametric Cox Model – an Application to Credit Risk.

5.1 Abstract

Banks and other institutions extend credit to customers, some of whom fail to repay as obliged and default. An accurate estimate of the probability of loan default is important for calculating minimum regulatory credit capital requirements and loan-loss provisions. Mixture-cure models have been used to estimate the probability of loan default, which supposes customers emanate from two distinct populations; (1) those susceptible to default; and (2) those not susceptible and hence "cured". While the Expectation Maximization (EM) algorithm is a common approach to estimate this model, it does not readily return estimates of either the baseline hazard for susceptible customers or variance estimates for model parameters without additional calculation steps.

This paper applies a constrained maximum likelihood algorithm to fit this model, where the constraint ensures the baseline hazard for susceptible customers is non-negative, jointly returning estimates for both regression parameters and the baseline hazard. This provides two key benefits: (1) an asymptotic variance matrix for all model parameters allowing joint inference on regression parameters and the baseline hazard without computationally intensive methods such as a bootstrap; and (2) a baseline hazard that is directly estimated without extra calculations using, for example, the Breslow (1972) method. Comparison of our method with the EM method is also provided.

Keywords: Mixture-Cure Survival Models, Constrained Maximum Likelihood Optimisation, Credit Risk Modelling.

5.2 Introduction

The primary business of banks and other credit-granting is lending funds to customers. Some customers may not make timely payments on their contractual obligations and default. An accurate estimate of the probability of default is a vital input to calculate both the institution's minimum regulatory credit capital (BIS, 2006) as well as its expected credit losses (IASB, 2014). These estimates also flow into wider risk-management activities of the bank, such as loan pricing, profit calculations and underwriting (Siddiqi, 2005).

Survival analysis has been readily applied to estimate default risk. These include singleevent survival analysis focussing on only the risk of credit default (see for example Banasik et al. (1999), Lessmann et al. (2015)) as well as competing risk analysis, recognising many customers successfully repay their loans (see for example Hall and Lundstedt (2005), Stepanova and Thomas (2002), Deng et al. (2000), Wycinka (2019), Thackham and Ma (2020b)). Another approach, mixture-cure analysis, assumes customers are from two distinct populations; (1) those susceptible to default; and (2) those not susceptible to default. This means the survival curve plateaus at a level of the long-run cure probability. These models were initially applied in studies of diseases, beginning with Berkson and Gage (1952), Farewell (1982) and Farewell (1986) who devised parametric approaches. Later extensions by authors such as Kuk and Chen (1992), Peng and Dear (2000) and Sy and Taylor (2000) extend the method to include the semi-parametric Cox model. Cai et al. (2012) implement this model via the smcure R package, with parameters estimated using the expectation-maximisation (EM) algorithm (Dempster et al., 1977).

Mixture-cure models have been applied to credit risk, recognising that not all customers default on their obligations, allowing not only if a customer is likely to default (their susceptibility) but also when they are likely to default (their survival time). Tong et al. (2012) outlines that mixture-cure models can be applied to estimate the risk of credit default by estimating a mixture-cure model using United Kingdom personal loan data using the EM algorithm. Dirick et al. (2017a) details how mixture cure models are highly applicable to credit risk modelling as the vast majority of loans do not default. Dirick et al (2016) extend this work to additionally include macroeconomic variables as discrete time-varying covariates to construct the full likelihood, and the EM algorithm to estimate model parameters. Alves and Dias (2015) apply a mixture-cure model to a sample of Portuguese car-loan data using four parametric baseline distributions for the susceptible population (exponential, Weibull, log-normal, log-logistic) and using maximum likelihood to estimate these fully-parametric models.

Kuk and Chen (1992) detail that the hazard of the semi-parametric mixture-cure model is no longer proportional to a function of time, so that estimating the model via partial likelihood is not possible. Thus mixture-cure survival models typically estimate parameters using the EM algorithm. This approach maximises a complete data likelihood, driven by the latency of the susceptible population. Some approaches, such as those introduced by Murphy (1994) and Murphy (1995) for frailty survival models, and adapted by Lu (2008) for mixture-cure models, estimate the non-parametric mixture cure model via maximum likelihood using the cumulative baseline hazard. Estimating the baseline hazard remains an area of research and is important to the mixturecure model and there have been several published approaches. Corbiere et al. (2009) estimates the mixture-cure model using penalised maximum likelihood, parametrising the cumulative hazard using cubic I-splines. Hua and Xiang (2013) cater for interval censoring using transformed baseline hazard function using splines. Andersson et al. (2011) fit a flexible parametric cure model using log-cumulative excess hazard of the relative risk, fitting their model in Stata with restrictive cubic splines. Patilea and VanKeilegom (2017) model the cure probably parametrically (such as via a logistic transform) but the survival component of the model non-parametrically, devising an inversion formula based on the cumulative baseline hazard. Liu and Shen (2009) develop a semi-parametric mixture-cure model for interval censored data, parametrising the cumulative baseline hazard using non-decreasing stepfunctions. Sy and Taylor (2000) employ the EM algorithm to estimate regression parameters and either the Breslow estimator for the baseline hazard or the Kaplan-Meier estimator for baseline survival. Corbiere and Joly (2007) parametrise the baseline survival function using a Weibull and exponential distributions.

In this paper, we outline a maximum likelihood approach to estimate the mixture-cure semi-parametric Cox model via constrained optimisation to jointly estimate the regression parameters and the baseline hazard for the susceptible population where the constraint respects the latter's non-negativity. This is achieved by approximating the baseline hazard using non-negative basis functions and applying an extended version of the multiplicative-iterative algorithm of Ma et al. (2014) (see also Thackham and Ma (2020a) and Thackham and Ma (2020b) for applications of this algorithm to credit risk). Our approach deals with two key aspects of estimating the mixture-cure survival model that remain unaddressed in the literature, these being production of:

- 1. a Hessian matrix for both the regression parameters and the baseline hazard, allowing joint inference without the need for a computational intensive method such as bootstrapping; and
- 2. a direct estimation of the baseline hazard for the susceptible population, constrained to respect its non-negativity, together with an estimate of its variance, without the need to undertake an additional step, such as the Breslow (1972) estimator.

We employ our method to estimate a mixture-cure semi-parametric Cox model using an anonymised sample of real home-loan data. We show our maximum likelihood method is able to estimate regression parameters with comparable accuracy to the EM algorithm. Specifically, our maximum-likelihood approach has two key advantages of the over EM algorithm: it does not rely on taking expected values of latent variables; and it readily returns an invertible Hessian so that asymptotic variance estimates can be produced, without resort to bootstrap. As our method readily returns asymptotic variance estimates we can provide z-statistics for drawing inferences (without the need for a bootstrap) on regression parameters. Additionally, as our method also readily returns an estimate of the baseline hazard along with asymptotic variance estimates, we are able to plot the baseline hazard with a 95% piecewise confidence intervals, confirming that the susceptible population has a humped-hazard.

This paper proceeds as follow. Section 5.3 outlines our methodology, including the likelihood function, gradient vector an Hessian matrix. Section 5.4 details the multiplicative-iterative algorithm used by our method for model estimation. We implement our model in R, which we use in section 5.5 to undertake both the simulation and the applied analysis. We conclude with a discussion in section 5.6. Our paper is supplemented by an appendix which provides additional detail of the multiplicative-iterative algorithm in section 5.4.

5.3 Maximum Likelihood Estimation

5.3.1 The Likelihood Function

This section begins by outlining notation used in this paper. Let R_i be a susceptibility indicator so that $R_i = 1$ if subject *i* is susceptible and $R_i = 0$ for otherwise. Let T_i be the time at which the event of interest occurs for the *i*th subject in the susceptible population, and C_i be the non-informative (independent) right censoring time for the *i*th subject in the susceptible population. So the observable survival time for all subjects is $Y_i = \min(T_i, C_i)$ for susceptible subjects and $Y_i = \infty$ for non-susceptible subjects, with y_i representing observed realisations. Thus each value of y_i can represent the time an event of interest occurs if $\delta_i = 1$, or when $\delta_i = 0$, this individual can be either right censored for the event of interest or belong to the non-susceptible population. To simplify discussions we let the combined survival time be

$$T_i^{\dagger} = \begin{cases} T_i, & \text{if } i^{th} \text{ subject is susceptible} \\ \infty, & \text{if } i^{th} \text{ subject is not susceptible.} \end{cases}$$
(5.1)

In equation (5.2) below we express the relationship between survival functions $S_i^{\dagger}(t) = P(T_i^{\dagger} > t)$ and $S_i(t) = P(T_i > t)$.

In addition, assume that for subject *i* we observed *p* covariates and their values are given in the vector $\mathbf{x}_i^T = [x_{i1}, \dots, x_{ip}]$ and *q* covariates in the vector $\mathbf{w}_i^T = [w_{i1}, \dots, w_{iq}]$ that help explain susceptibility. Defining $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ as regression coefficients associated with \mathbf{x}_i^T and \mathbf{w}_i^T respectively, and $h_0(t)$ as the non-parametric continuous baseline hazard for the susceptible population, the mixture cure model is

$$S_{i}^{\dagger}(t|\boldsymbol{w}_{i},\boldsymbol{x}_{i}) = \pi_{i}(\boldsymbol{w}_{i})S_{i}(t|\boldsymbol{R}_{i}=1,\boldsymbol{x}_{i}) + 1 - \pi_{i}(\boldsymbol{w}_{i})$$
(5.2)

where, in this paper we consider the following specifications:

$$\pi_i(\boldsymbol{w}_i) = 1/(1 + \exp(-\boldsymbol{w}_i^T \boldsymbol{\alpha}))$$
(5.3)

$$h_i(t|R_i = 1, \mathbf{x}_i) = h_0(t)e^{\mathbf{x}_i^T \boldsymbol{\beta}}$$
(5.4)

$$S_i(t|R_i = 1, \boldsymbol{x}_i) = e^{-\int_0^t h_i(\xi|R_i = 1, \boldsymbol{x}_i)d\xi}$$
(5.5)

where $\pi_i(\mathbf{w}_i)$ is the probability of subjects in the susceptible population. Here, the function $S_i^{\dagger}(t|\mathbf{w}_i, \mathbf{x}_i)$ has the property that it "plateaus" at the long-run cure rate (ie: as $t \to \infty$ then $S_i^{\dagger}(t|\mathbf{w}_i, \mathbf{x}_i) \to 1 - \pi_i(\mathbf{w}_i)$). When there is no non-susceptible population (ie: $\pi_i(\mathbf{w}_i) = 1$ for all subjects), equation (5.2) becomes the same as the conventional survival function (ie: as $t \to \infty$ then $S^{\dagger}(t) \to 0$).

For the hazard in equation (5.4) to be valid, we must constrain $h_0(t) \ge 0$ in our model, which this paper does when developing the constrained maximum likelihood algorithm to fit our model. However as $h_0(t)$ is an infinite dimensional parameter, estimating $h_0(t)$ using a finite number of observations without restrictions is infeasible. A common strategy is to simplify $h_0(t)$ to a finite dimensional subspace where its dimension grows with the sample size *n* but at a slower rate. We require that when $n \to \infty$ the simplified $h_0(t)$ converges to the true $h_0(t)$ (Wong and Severini, 1991). The subspace we employ has the dimension $m (\leq n)$ and has non-negative basis functions $\psi_u(t)$ (where u = 1, ..., m) such that

$$h_0(t) := \sum_{u=1}^m \theta_u \psi_u(t),$$
(5.6)

where $\psi_u(t) \ge 0$ are non-negative basis functions. Corresponding cumulative baseline hazard is given by $H_0 = \sum_{u=1}^m \theta_u \Psi_u(t)$, where $\Psi_u(t) = \int_{s=0}^t \psi_u(s) ds$ is the cumulative basis function. Together, this implies the cumulative hazard

$$H_i(t) = H_0(t)e^{\mathbf{x}_i^T\boldsymbol{\beta}}.$$
(5.7)

The available data to estimate the model are $(y_i, \delta_i, w_i, x_i)$ for i = 1, ..., n. Let θ be the *m* vector for all θ_u , so that the likelihood of the mixture-cure model for all *i* independent subjects is

$$L(\alpha, \beta, \theta) = \prod_{i=1}^{n} \left[\pi_i(w_i) f_i(y_i | R_i = 1, x_i) \right]^{\delta_i} \left[\left(1 - \pi_i(w_i) \right) + \pi_i(w_i) S_i(y_i | R_i = 1, x_i) \right]^{1 - \delta_i}$$
(5.8)

where $f_i(y_i|R_i = 1, x_i) = h_i(y_i|R_i = 1, x_i) \times S_i(y_i|R_i = 1, x_i)$ is the density function for the susceptible population. Note that when all subjects are susceptible (ie: $\pi(w_i) = 1$ for all subjects) then the likelihood in equation (5.8) reduces to the likelihood for the semiparametric Cox model. For notational ease, the conditioning on R_i , x_i and w_i are dropped, and the corresponding log-likelihood is

$$l(\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\theta}) = \sum_{i=1}^{n} \delta_{i} \ln \left[\pi_{i}\right] + \delta_{i} \ln \left[h_{0}(y_{i})\right] + \delta_{i} \left[\boldsymbol{x}_{i}^{T} \boldsymbol{\beta}\right] - \delta_{i} \left[H_{i}(y_{i})\right] + (1 - \delta_{i}) \ln \left[\pi_{i} S_{i}(y_{i}) + (1 - \pi_{i})\right]$$
(5.9)

where we need to estimate the vector of parameters $\boldsymbol{\eta} = [\boldsymbol{\alpha}^T, \boldsymbol{\beta}^T, \boldsymbol{\theta}^T]^T$.

5.3.2 Gradient Vector and Hessian Matrix

Our algorithm to estimate the model parameters requires first and second derivatives of the log-likelihood in equation (5.9) with respect to η . We note that:

$$\frac{\partial}{\partial \alpha_j} \pi_i = w_{ij} \pi_i^2 e^{-w_i^T \alpha} = -\frac{\partial}{\partial \alpha_j} (1 - \pi_i) = w_{ij} \pi_i (1 - \pi_i)$$
(5.10)

and further the following expressions which are useful in the calculation of the gradient and Hessian matrices: = (S(x) - 1)

$$b_i = \frac{\pi_i (S_i(y_i) - 1)}{\pi_i S_i(y_i) + 1 - \pi_i}$$
(5.11)

$$d_{i} = \frac{\pi_{i} S_{i}(y_{i})}{\pi_{i} S_{i}(y_{i}) + 1 - \pi_{i}}$$
(5.12)

The first derivatives are

$$\frac{\partial l}{\partial \alpha_j} = \sum_{i=1}^n \left[\delta_i + (1-\delta)b_i \right] (1-\pi_i) w_{ij}$$
(5.13)

$$\frac{\partial l}{\partial \beta_j} \sum_{i=1}^n \left[\delta_i - \left(\delta_i + (1 - \delta_i) d_i \right) H_i(y_i) \right] x_{ij}$$
(5.14)

$$\frac{\partial l}{\partial \theta_u} = \sum_{i=1}^n \left[\delta_i \frac{\psi_u(y_i)}{h_0(y_i)} - \left(\delta_i + (1 - \delta_i) d_i \right) \Psi_u(y_i) e^{\mathbf{x}_i^T \mathbf{\beta}} \right].$$
(5.15)

Elements of the Hessian matrix are

$$\frac{\partial^2 l}{\partial \alpha_j \partial \alpha_k} = -\sum_{i=1}^n \left[\delta_i - (1 - \delta_i)(1 - b_i) \right] (1 - \pi_i) b_i (1 - b_i) w_{ij} w_{ik}$$
(5.16)

$$\frac{\partial^2 l}{\partial \beta_j \beta_k} = -\sum_{i=1}^n \left[\delta_i + (1 - \delta_i) d_i \left(1 - (1 - d_i) H_i(y_i) \right) \right] H_i(y_i) x_{ij} x_{ik}$$
(5.17)

$$\frac{\partial^2 l}{\partial \theta_u \theta_v} = -\sum_{i=1}^n \left[\delta_i \frac{\psi_u(y_i)\psi_v(y_i)}{h_0^2(y_i)} - (1 - \delta_i)d_i(1 - d_i)\Psi_u(y_i)\Psi_v(y_i)e^{2x_i^T \beta} \right]$$
(5.18)

$$\frac{\partial^2 l}{\partial \beta_j \alpha_k} = -\sum_{i=1}^n (1 - \delta_i) d_i (1 - d_i) H_i(y_i) x_{ij} w_{ik}$$
(5.19)

$$\frac{\partial^2 l}{\partial \alpha_j \theta_u} = -\sum_{i=1}^n (1 - \delta_i) d_i (1 - d_i) \Psi_u(y_i) w_{ij}$$
(5.20)

$$\frac{\partial^2 l}{\partial \beta_j \theta_u} = -\left[\delta_i + (1 - \delta_i)d_i \left(1 - (1 - d_i)\right)H_i(y_i)\right] x_{ij} \Psi_u(y_i)^{\mathbf{x}_i^T \boldsymbol{\beta}}.$$
(5.21)

5.4 Constrained Optimisation Algorithm

The Karush–Kuhn–Tucker (KKT) (Karush (1939), Kuhn and Tucker (1951)) first-order necessary conditions for the constrained optimal solution of β , α and θ are

$$\frac{\partial l}{\partial \beta_j} = 0, \tag{5.22}$$

$$\frac{\partial l}{\partial \alpha_j} = 0, \tag{5.23}$$

$$\frac{\partial l}{\partial \theta_u} = 0 \text{ if } \theta_u > 0 \text{ or } \frac{\partial l}{\partial \theta_u} < 0 \text{ if } \theta_u = 0$$
 (5.24)

Some algorithms for constrained optimisation can be found in, for example, Luenberger and Ye (2008). However, many of them are less efficient for our problem when m is large. We therefore instead develop in this section an algorithm which is efficient for a large m.

We adopt the following strategy to solve equations given in (5.22), (5.23) and (5.24). Beginning with estimates $\beta^{(k)}$, $\alpha^{(k)}$ and $\theta^{(k)}$ at iteration k, iteration k + 1 comprises the following alternating steps.

- 1: Compute $\boldsymbol{\beta}^{(k+1)}$ so that $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{\theta}^{(k)}) \ge l(\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{\theta}^{(k)})$.
- **2:** Compute $\boldsymbol{\alpha}^{(k+1)}$ so that $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \ge l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{\theta}^{(k)}).$
- 3: Compute $\boldsymbol{\theta}^{(k+1)} \ge 0$ so that $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\theta}^{(k+1)}) \ge l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\theta}^{(k)}).$

These incremental conditions assure $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\theta}^{(k+1)}) \ge l(\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{\theta}^{(k)})$ at the end of iteration k + 1, which is a key requirement for convergence of this algorithm. Steps 1 and 2

are solved by the Newton algorithm incorporating line search steps, and Step 3 is solved by a multiplicative-iterative (MI) algorithm (e.g. Chan and Ma (2012) and Ma (2010)) designed to respect the non-negative constraints on θ . We call this algorithm the Newton-MI algorithm similar to Ma et al. (2014), Thackham and Ma (2020a), and Thackham and Ma (2020b).

To update β , we employ one iteration of the Newton algorithm with line search. Starting with $\beta^{(k)}$ and using a line-search with step size $\omega_1^{(k)} \in (0, 1]$, we have

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} - \omega_1^{(k)} (\mathbf{X}^T \mathbf{B}_2^{(k)} \mathbf{X})^{-1} \mathbf{X}^T (\mathbf{B}_1^{(k)} \mathbf{1}_n + \boldsymbol{\delta}), \qquad (5.25)$$

where **X** (of size $n \times p$) is the model matrix of covariates and its *i*-th row is given by \mathbf{x}_i^T , \mathbf{B}_1 and \mathbf{B}_2 are a diagonal matrix whose ii^{th} element are detailed in appendix A, $\mathbf{1}_n$ is an *n*-vector of 1's and $\boldsymbol{\delta}$ is an *n*-vector for δ_i . Matrices $\mathbf{B}_1^{(k)}$ and $\mathbf{B}_2^{(k)}$ are \mathbf{B}_1 and \mathbf{B}_2 respectively with $\boldsymbol{\beta} = \boldsymbol{\beta}^{(k)}$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(k)}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$. Matrix $\mathbf{X}^T \mathbf{B}_2 \mathbf{X}$ is the negative Hessian of $l(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\beta}$. The line search parameter $\omega_1^{(k)}$ helps to achieve $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{\theta}^{(k)}) \ge l(\boldsymbol{\beta}^{(k)}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{\theta}^{(k)})$.

Similarly, the Newton method with line search is also applied to update the logistic regression parameter α . Starting with $\alpha^{(k)}$ and using a line-search with step size $\omega_2^{(k)} \in (0, 1]$, we have

$$\boldsymbol{\alpha}^{(k+1)} = \boldsymbol{\alpha}^{(k)} - \omega_2^{(k)} (\mathbf{W}^T \mathbf{A}_2^{(k)} \mathbf{W})^{-1} \mathbf{W}^T (\mathbf{A}_1^{(k)} \mathbf{1}_n + \boldsymbol{\xi}),$$
(5.26)

where **W** (of size $n \times q$) is the model matrix of covariates for the logistic regression and its *i*-th row is given by \mathbf{w}_i^T , \mathbf{A}_1 and \mathbf{A}_2 are a diagonal matrix whose ii^{th} element are detailed in appendix A, and $\boldsymbol{\xi}$ is an *n*-vector for $\delta_i(1 - \pi_i)$. Matrices $\mathbf{A}_1^{(k)}$ and $\mathbf{A}_2^{(k)}$ are \mathbf{A}_1 and \mathbf{A}_2 respectively with $\boldsymbol{\beta} = \boldsymbol{\beta}^{(k+1)}$, $\boldsymbol{\alpha} = \boldsymbol{\alpha}^{(k)}$ and $\boldsymbol{\theta} = \boldsymbol{\theta}^{(k)}$. Matrix $\mathbf{W}^T \mathbf{A}_2 \mathbf{W}$ is the negative Hessian of $l(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{\theta})$ with respect to $\boldsymbol{\alpha}$. The line search parameter $\omega_2^{(k)}$ helps to achieve $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\theta}^{(k)}) \ge l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k)}, \boldsymbol{\theta}^{(k)})$.

Finally for updating θ , we adopt the multiplicative-iterative (MI) algorithm which respects the non-negative constraint on θ . Let both **C** and **C**^{*} be $n \times m$ matrices but their (i, u)elements are $\psi_u(y_i)$ and $\Psi_u(y_i)$ respectively. Let δ be the *n*-vector for δ_i and **f** the *n*-vector for $\delta_i e^{\mathbf{x}_i^T \boldsymbol{\beta}} + (1 - \delta_1) e^{\mathbf{x}_i^T \boldsymbol{\beta}} d_i$. The MI algorithm updates θ according to

$$\boldsymbol{\theta}^{(k+1)} = \boldsymbol{\theta}^{(k)} + \omega_3^{(k)} \mathbf{S}^{(k)} \left(\mathbf{C}^T [\mathbf{D}^{(k)}]^{-1} \boldsymbol{\delta} - [\mathbf{C}^*]^T \mathbf{f}^{(k)} \right),$$
(5.27)

where **D** and **S** are diagonal matrices with elements $h_0(y_i)$ and $\theta_u/(\mathbf{C}^{*T}\mathbf{f} + \varepsilon)$ respectively, and here ε is a small threshold used to avoid the corresponding denominator being zero. The step size $\omega_3^{(k)} \in (0, 1]$ again guarantees that $l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\theta}^{(k+1)}) \ge l(\boldsymbol{\beta}^{(k+1)}, \boldsymbol{\alpha}^{(k+1)}, \boldsymbol{\theta}^{(k)})$.

All the above line searches can be efficiently conducted using, for example, the Armijo rule (Luenberger and Ye, 2008). When $\mathbf{B}_2^{1/2}\mathbf{X}$ and $\mathbf{A}_2^{1/2}\mathbf{W}$ have full column rank, matrices $\mathbf{X}^T\mathbf{B}_2\mathbf{X}$ and $\mathbf{W}^T\mathbf{A}_2\mathbf{W}$ are positive definite so that the updates for $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ are well defined. Following the same argument as in Chan and Ma (2012) we can show that (i) if $\boldsymbol{\theta}^{(k)}$ is non-negative then $\boldsymbol{\theta}^{(k+1)}$ is also non-negative, and (ii) under certain regularity conditions, this algorithm converges to a solution satisfying the KKT conditions.

Regarding implementation, the user may parametrise the baseline hazard θ by specifying the number of observed events within each piece-wise constant interval. The number of knots is usually taken as the cubic root of the number of events.

5.5 Results

In this section, we demonstrate our method using both a simulation study as well as an applied example to an anonymised real-world credit risk dataset. These are benchmarked against the smcure R package (Chao et al., 2012)

5.5.1 Simulation Results

For our simulation study, we draw M = 1,000 survival times across two different samples sizes (n = 100 and n = 1,000) from a mixture-cure model specified by (5.2) – (5.5), where the covariates are generated according to $w_1 \sim N(0,1)$ and $x_1 \sim N(0,1)$. Bender et al. (2005) details that survival times for the susceptible subjects can be drawn by inverting the cumulative hazard H(t) such that $T = H^{-1}[-\log(u)]$ where $u \sim Unif(0,1)$. For all subjects, the independent right-censoring time is drawn from a uniform distribution with lower bound of zero and upper bound is chosen to approximate the desired censoring proportion (either 20% or 50%). Susceptibility is simulated by drawing *Bernouli*(π) for each subject and if this value is 1 the survival time is set to the censoring time. Parameters are set to be: $\alpha_0 = 1, \alpha_1 = 1, \beta_1 = 1, \lambda = 1, \nu = 1$.

		N cen	s.= 50%	N cen	f = 100 s.= 20%	N cen	= 2000 s.= 50%	N = cens	= 2000 = 20%
Parm	Measure	EM Algo.	Constrained ML	EM Algo.	Constrained ML	EM Algo.	Constrained ML	EM Algo.	Constrained ML
\hat{lpha}_0	Bias SD MSE	0.6182 1.0210 1.4246	0.2176 0.8053 0.6959	-0.3061 0.3895 0.2454	$0.0562 \\ 0.2564 \\ 0.0689$	0.7769 0.1272 0.6197	0.6433 0.122 0.4287	-0.1782 0.1053 0.0428	0.0043 0.0874 0.0077
$\hat{\alpha}_1$	Bias SD MSE	0.7156 0.8758 1.2791	$\begin{array}{c} 0.5138 \\ 0.6093 \\ 0.6353 \end{array}$	-0.1249 0.3972 0.1734	$\begin{array}{c} 0.0919 \\ 0.2443 \\ 0.0681 \end{array}$	0.2862 0.116 0.0954	$0.2375 \\ 0.1103 \\ 0.0686$	-0.0992 0.0886 0.0177	-0.0045 0.0856 0.0073
\hat{eta}_1	Bias SD MSE	-0.2180 0.2253 0.0983	-0.2136 0.1861 0.0803	-0.0882 0.1725 0.0375	$\begin{array}{c} 0.0240 \\ 0.1638 \\ 0.0274 \end{array}$	-0.1476 0.0534 0.0246	-0.1291 0.0501 0.0193	-0.0428 0.054 0.0047	-0.0032 0.0488 0.0024

Table 5.1: Simulation Results of Parameters Estimates using Expectation Maximisation (EM) and Constrained Maximum Likelihood (ML)

The results for the bias, standard deviation (SD) and mean-square error (MSE) across sample-size and censoring proportion from the simulation are in table 5.1. The results show our maximum likelihood approach returns comparable accuracy for coefficients in the survival component (β_1) and lower bias for coefficients in the logistic component (α_1 and α_0). Not only does our constrained maximum likelihood algorithm return estimates for regression parameters, but it also return (in the same optimisation) estimates for the baseline hazard for the susceptible population. Figure 5.1 demonstrates for each simulation setting that our constrained maximum likelihood can capture the true baseline hazard, along with the 95% empirical confidence intervals.



Figure 5.1: Comparison of true and Maximum Likelihood estimated baseline hazards, together with simultaneous 95% CIs, using Constrained Maximum Likelihood (ML)

5.5.2 Application to Credit Risk Data

We apply our maximum likelihood estimation of the mixture-cure model to a real-world credit risk dataset comprising a randomised and anonymised sample of n = 100,000 Australian home loans spanning the calendar years 2003 to 2014. The event of interest for this data is the default at any time during the life for each home loan. Table 5.2 details the number of loans which have observed default events.

Outcome for Loan	Count	Percent
Observed Default (event of interest) No Default Observed (either censored or non-susceptible)	1,823 98,177	1.823% 98.177%
Total	100,000	100.000%

Table 5.2: Number of Defaults and Observations in the Credit Risk Dataset

The data are sourced from a large Australian mortgage originating bank. The data are proprietary and commercially sensitive (thus not available publicly), and has been provided to one of the author's to help demonstrate the applicability of the methodology in this paper. Given the paper is methodological, the focus is not the application which is simply a demonstration of the method.

The data contains 11 baseline covariates as per table 5.3. We use this dataset to apply our maximum likelihood estimation method to fit the mixture cure model and compare this to the EM algorithm. The baseline cause-specific baseline hazard for the susceptible population is parametrised using 7 piece-wise constant indicator functions, with knots are selected so that there are roughly an equal number of observed events within each piece-wise constant interval.

#	Covariate	Description (* = mean corrected)
1 2 3 4 5 6 7 8 9 10 11	Mortgage Insured Credit Card Personal Loan Repayment Method Repayment Frequency Loan Term Borrower Tenure* Borrower Age* Total Home Loans* Opening Balance* Original Loan to Value Ratio*	Does the home loan have lenders mortgagee insurance? (Yes / No)? Do the borrowers have a credit card? (Yes/No) Do the borrowers have a personal loan? (Yes/No) Are repayments principal and interest (P&I) or interest only (IO)? Are repayments made monthly or fortnightly/other What is the original loan term (1-5 years or 6+ years)? How many months has the borrower been a customer (0,1,2)? How old is the primary borrower (years)? Total number of home loans the borrowers have (integer 1, 2,) The original balance when the home loan opened (\$millions) The original loan balance divided by the original the value of the home securing the loan.

Table 5.3: Description of the 11 covariates in the Home Loan dataset

and survival components of the mixture cure model. The first three columns of the table details the model component (logistic or survival) and the covariate. The final four columns of the table show the results of the fitted mixture-cure model. Column (A) displays the parameter estimates returned via the EM algorithm while column (B) displays the parameter estimates returned via our constrained maximum likelihood algorithm. In addition, column (C) displays the asymptotic standard error that is readily returned from our constrained maximum likelihood approach which we use in columns (D) and (E) to calculate a z-statistic and associated p-value, concluding all effects entered into the model are significant. Note that these inferences are able to be drawn without use of computationally intensive bootstraping, but instead are the asymptotic variance estimates form the constrained maximum likelihood estimation from the diagonals of the inverse Hessian matrix.

Applying our constrained maximum likelihood approach to this data results in intuitive parameter estimates. For susceptibility, customers who required mortgagee insurance coverage or those with shorter tenure are more likely to be susceptible of credit default. These risk signals are intuitive, as mortgage insurance is typically required for high loan-to-value loans and customers with a shorter length of relationship with a bank tend to be higher default risk.

For survival time, covariates can be grouped into four groups. The first relates to details of the home loan granted to the customer. Customers whose home loans had a higher opening balance, higher loan-to-value ratio, or longer initial term have an increased time to credit default. The second group relate to other products the customer also holds in addition to their home loan. Customers with multiple home loans or at least one credit card are lower credit default risk, suggestive that these customers already have a long-term and multi-product relationships with the bank. Customers with at least one personal loan with the bank are higher risk. The third group relate to how the customer plans to repay their loan. Customers who repay fortnightly (rather than monthly) are lower risk. This makes sense, as these customers pay more money to the bank sooner, as well as the fact that they make the amount of money being paid in future more susceptible to rises in interest rates. The fourth group is customer age, with older customers being riskier, which for this data may be detecting a refinancing risk.

PAPER 3: Maximum L	Likelihood Estimation of	of the Mixture	Cure Semi-Para	metric Cox
110		Model – an	Application to	Credit Risk

			(A)	(B)	(C)	(D)	(E)
			EM	ML	ML	ML	ML
Component	Covariate	Level	Estimate	Estimate	Std Err	Z	p-val
Logistic	Intercept		-2.8493	-2.8495	0.3139	9.0777	< .0001
Logistic Logistic	Mortgage Insured	Yes No	0.6803	0.6799 0	0.0726	9.3650	< .0001
Logistic	Borrower Tenure		-0.0023	-0.0024	0.0003	8.0000	< .0001
Survival Survival	Credit Card	Yes No	-0.4029	-0.3943	0.045	8.7622	< .0001
Survival	Personal Loan	Yes	0.7054	0.6945	0.0860	8.0756	< .0001
Survival	Repayment Method	PI	0.1510	0.1484	0.0577	2.5719	0.0292
Survival	Repayment Frequency	Fortnight	0.2852	0.2603	0.0522	4.9866	< .0001
Survival Survival Survival Survival Survival	Total Home Loans Opening Balance Original Loan to Value Borrower Age	Other	-0.2068 0.4408 0.0156 0.0127	-0.1958 0.4248 0.0154 0.0128	0.0172 0.1019 0.0016 0.0020	11.384 4.1688 9.6250 6.4000	< .0001 < .0001 < .0001 < .0001
Survival Survival	Loan Term	1-5 years 6+ years	-0.2886 0	-0.2996 0	0.0583	5.1389	< .0001

 Table 5.4: Results of Parameters Estimates using Expectation Maximisation (EM) and

 Analytical Maximum Likelihood (ML)

Our maximum likelihood method also readily returns an estimate of the baseline hazard, along with an asymptotic variance. Figure 5.2 plots the baseline hazard for the susceptible population, and uses the asymptotic variance to plot the 95% piecewise confidence interval derived from inverting the Hessian matrix from our maximum likelihood approach. The plot confirms that the hazard for the risk of default for the susceptible population is "humped", so that for this data the risk of default rises to a peak before starting to fall. Despite often varying in size and timing betweens different countries and different asset classes, this effect is well documented (see for example Im et al. (2012), Bellotti and Crook (2013), Thackham and Ma (2020a) and Thackham and Ma (2020b)). A benefit of applying our constrained maximum likelihood estimation approach to credit risk data is that it provides clearer and easy interpretation of this effect, together with the ability to discern its significance using the asymptotic variance our method produces.



Figure 5.2: Baseline Hazard and Piecewise 95% Confidence Interval for the Susceptible Population

Table 5.5 outlines calibration and discrimination results for each model. The first column is the time since origination while the second column gives the observed cumulative number of defaults up to time T. The third and fourth columns respectively give the expected number of defaults from the Constrained Maximum Likelihood (ML) and Expectation Maximisation (EM) estimation approach. The expected number of defaults up until time T needs to recognise the competing risk of redemption, so the expression is multiplied by $K_e(t)$, the Kaplan-Meier estimator (Kaplan and Meier, 1958) of early loan repayment at time t. Following the approach of Tong et al. (2012), this is achieved using

$$E[D(T)] = \sum_{i=1}^{n} \sum_{t=1}^{T} \left(S_i(t-1) - S_i(t) \right) K_e(t)$$
(5.28)

where $S_i(t)$ is the population survival estimate at time *t* for subject *i*. The final two columns of table 5.5 are the AUC statistics, together with 95% confidence intervals. The AUC could be improved by extending the methodology to include time-varying covariates, however this is not addressed in this paper and is instead left as an avenue of future research.

PAPER 3: Maximum Likelihood Estimation of the Mixture Cure Semi-Parametric Cox 112 Model – an Application to Credit Risk

Time	Observed	Expected	Expected	AUC (95%CI)	AUC (95%CD)
(monthe)	Defeulte	Defaulte MI	Defeulte EI	MU	
(monuis)	Delauits	Defaults ML	Delaults EL	IVIL	EL
12	313	351	243	0 7334 (0 7077 0 7592)	0 7337 (0 7080 0 7594)
12	70	700	213	0.7551(0.7077, 0.7552)	0.7557 (0.7000, 0.7571)
24	/64	128	657	0./145(0.09/3, 0./31/)	0./149(0.69//, 0./321)
36	1,099	1.031	983	0.7107 (0.6959, 0.7254)	0.7106 (0.6958, 0.7253)
48	1,376	1,303	1,280	0.7025 (0.6889, 0.7160)	0.7021 (0.6885, 0.7156)
60	1,570	1,519	1,516	0.6995 (0.6867, 0.7122)	0.6988 (0.6860, 0.7115)
72	1.694	1,679	1.710	0.6978 (0.6855, 0.7102)	0.6968 (0.6845, 0.7092)
84	1.754	1.808	1.833	0.6950 (0.6828, 0.7072)	0.6936 (0.6814, 0.7057)
96	1,796	1,908	1,955	0.6925 (0.6804, 0.7046)	0.6901 (0.6780, 0.7022)
108	1.814	1,987	2.038	0.6914 (0.6793, 0.7034)	0.6881 (0.6761, 0.7001)
120	1,022	2,020	2,083	0 6008 (0 6787 0 7028)	0.6969 (0.6749 0.6099)
120	1,025	2,050	2,065	0.0908(0.0787, 0.7028)	0.0808 (0.0748, 0.0988)

 Table 5.5: Calibration Results for Expectation Maximisation (EM) and Constrained

 Maximum Likelihood (ML)

5.6 Conclusion

Financiers critically require accurate estimates of the probability of default for loans they grant to customers, for use as inputs to both minimum regulatory credit capital requirements and loan-loss provisioning. These estimates also flow into wider risk-management activities of the bank, such as loan pricing, profit calculations and underwriting (Siddiqi, 2005).

Many methods have been deployed to estimate probability of default, including mixturecure models which suppose customers emanate from two distinct populations; (1) those susceptible to default; and (2) those not susceptible to default and hence "cured". Under this model, covariates can be used to explain not only loan default susceptibility but also loan default survival.

While the commonly applied approach to use the Expectation Maximization algorithm of Dempster et al. (1977) provides estimates of regression coefficients, it does not readily return estimates of either the baseline hazard for the susceptible population or the estimates of variances for model parameters without resort to additional calculation steps, such as bootstrapping to recover variance estimates for regression parameters, or recover baseline hazard estimates.

In this paper, we apply a constrained maximum likelihood approach to estimate the mixture-cure model. Using a constrained optimisation, we jointly estimate both regression parameters and the baseline hazard where the constraint ensures the hazard's non-negativity. This method applies an extended version of the multiplicative-iterative algorithm of Ma

et al. (2014) which implements a non-negative basis function for the baseline hazard. Our approach generates two key benefits, being: (1) provision of an invertible Hessian matrix for all parameters, so that joint inference on regression parameters and the baseline are possible without a bootstrap; and (2) a estimate of the baseline hazard that is not estimated via (for example) a logarithm transform or an extra calculation step (for example) using the Breslow (1972) estimator.

When our maximum likelihood estimation method of the mixture-cure model is applied to an anonymised sample of real home-loan data, results show comparable accuracy to the EM algorithm. Via the readily returned asymptotic variance estimates, we calculate z-statistics, concluding regression parameters for this data are significant, without the need for a bootstrap. We additionally confirm using our method's estimate of the baseline hazard for the susceptible population is humped, and accompany this with by a 95% confidence interval calculated using the asymptotic variance returned from our constrained maximum likelihood approach.

Future avenues of research for our mixture-cure methodology could include extensions to cater for competing risks as well as for including time-varying covariates.

5.7 Appendix A

As referred to in section (4.6), details of the matrices in the constrained optimisation are as follows.

For the update to $\boldsymbol{\beta}$, \mathbf{B}_1 and \mathbf{B}_2 are diagonal matrices whose ii^{th} elements are given by $\mathbf{B}_1 = \operatorname{diag}\left(\left[\delta_i - \left(\delta_i + (1 - \delta_i)P_i\right)H_i(y_i)\right]\right)$, and $\mathbf{B}_2 = \operatorname{diag}\left(\left[\delta_i + (1 - \delta_i)P_i\left(1 - (1 - P_i)H_i(y_i)\right)\right]H_i(y_i)\right)$.

To ensure that the negative of the Hessian matrix is positive-definite, for the update to α , we use the following \mathbf{A}_1 and \mathbf{A}_2 are diagonal matrices whose ii^{th} elements are given by $\mathbf{A}_1 = \text{diag}\left(\left[\delta_i + (1-\delta)b_i\right](1-\pi_i)\right)$, and

$$\mathbf{A}_2 = \operatorname{diag}\left(\left[\delta_i - (1 - \delta_i)(1 - b_i)\right](1 - \pi_i)b_i(1 - b_i)\right)$$

References

- Alves, B. and Dias, J. (2015) Survival mixture models in behavioral scoring *Expert Systems* with Applications, 42(8), 3902–3910. https://doi.org/10.1016/j.eswa.2014.12.036
- Anderson, R. (20074). The Credit Scoring Toolkit Oxford.
- Austin, P. (2012). Generating survival times to simulate cox proportional hazards models with time-varying covariates. *Statistics in Medicine*, 31(29), 3946-3958. https://doi.org/10.1002/sim.5452
- Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54(6), 627-635. https://doi.org/10.1057/palgrave.jors.2601545
- Banasik, J., Crook, J., Thomas, L. (1999). Not if but when will borrowers default. *Journal* of the Operational Research Society, 50(12), 1185-1190. https://doi.org/10.2307/3010627
- Bellotti, T., and Crook, J. (2013). Forecasting and stress testing credit card default using dynamic models. International Journal of Forecasting, 29(4), 563-574. https://doi.org/10.1016/j.ijforecast.2013.04.003
- Bender, R., Augustin, T., and Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. *Statistics in Medicine*, 24(11), 1713-1723. https://doi.org/10.1002/sim.2059
- Berkson, Y. and Gage, R. (1952). Survival Curve for Cancer Patients Following Treatment. J. Amer. Statist. Assoc., 47(259), 501-515. https://www.jstor.org/stable/2281318

- BIS (2006). International Convergence of Capital Measurement and Capital Standards A Revised Framework Comprehensive Version *Bank for International Settlements*. Basel Framework
- Breslow, N. E. (1972). Contribution to the discussion of paper by D.R. Cox. *Journal of the Royal Statistical Society: Series B*, 34, 216–217 https://www.jstor.org/stable/2985181
- Cai, C., Zou, T., Peng, Y., and Zhang, J. (2012). smcure: An R-Package for Estimating Semiparametric Mixture Cure Models. *Computer Methods and Programs in Biomedicine*, 108(3), 1255-1260. https://doi.org/10.1016/j.cmpb.2012.08.013
- Cai, C., Zou, Y., Peng, Y., Zhang, J. (2012). smcure: Fit Semiparametric Mixture Cure Models *R package version 2.0*, https://CRAN.R-project.org/package=smcure
- Chan, R. H. and Ma, J. (2012). A multiplicative iterative algorithm for box-constrained penalized likelihood image restoration. *IEEE Trans. Image Processing*, 21(7), 3168-3181. https://ieeexplore.ieee.org/document/6156784
- Clapp, J., An, X., and Deng, Y. (2005). smcure: Unobserved Heterogeneity in Models of Competing Mortgage Termination. Social Science Research Network. https://ssrn.com/abstract=512624.
- Corbiere, F., Commenges, D., Taylor, J., Joly, P. (2009). A penalized likelihood approach for mixture cure models *Statistics in Medicine*, 28(3), 510–524. https://doi.org/10.1002/sim.3481
- Corbiere, F., Joly, P. (2007). A SAS macro for parametric and semiparametric mixture cure models *Computer Methods and Programs in Biomedicine*, 85, 173–180. https://doi.org/10.1016/j.cmpb.2006.10.008
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), 1-38. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x
- Deng, Y., Quigley, J., Van Order, R. (2000) Mortgage termination heterogeneity and the exercise of mortgage options. Econometrica, 68(2), 275-307. https://doi.org/10.1111/1468-0262.00110

- Dirick, L., Claeskens, G., and Baesens, V. (2017). Time to default in credit scoring using survival analysis: a benchmark study. *Journal of the Operational Research Society*, 68(6), 252-665. https://doi.org/10.1057/s41274-016-0128-9
- Dirick, L. Bellotti, T., Claeskens, G., Baesens, B. (2016). Macro-economic factors in credit risk calculations: Including time-varying covariates in mixture cure models. *Journal of Business and Economic Statistics*, 37(1), 40-53. https://doi.org/10.1080/07350015.2016.1260471
- Farewell, V. (1982). The Use of Mixture Models for the Analysis of Survival Data With Long-Term Survivors. *Biometrics*, 38(4), 1041-1046. https://www.jstor.org/stable/2529885
- Farewell, V. (1986). Mixture models in survival analysis: are they worth the risk? *Canadian Journal of Statistics*, 14(3), 257–262. https://www.jstor.org/stable/3314804
- Hall, A. and Lundstedt, K. (2005). The Competing Risks Framework for Mortgages: Modeling the Interaction of Prepayment and Default *Risk Management Association Journal*, September. https://tinyurl.com/yxhptgqv
- Hua, T. and Xiang, L. (2013). Efficient Estimation for Semiparametric Cure Models with Interval-Censored Data *Journal of Multivariate Analysis*, 121, 139-151. https://doi.org/10.1016/j.jmva.2013.06.006
- IASB. (2014). International Financial Reporting Standard 9. https://www.ifrs.org/issuedstandards/list-of-standards/ifrs-9-financial-instruments/
- Im, J., Apley, D., Qi, C., Shan, X. (2012) A time-dependent proportional hazards survival model for credit risk analysis. *Journal of the Operational Research Society*, 63(3), 306-321. https://doi.org/10.1057/jors.2011.34
- Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. [Master's thesis, M.Sc. Dissertation. Dept. of Mathematics], Univ. of Chicago.
- Kuhn, H. and Tucker (1951). Nonlinear programming. *Proceedings of 2nd Berkeley Symposium. Berkeley*, pp481-492.
- Kuk, A. and Chen, C. (1992). A Mixture Model Combining Logistic Regression With Proportional Hazards Regression. *Biometrika*, 79:531-541.

- Lessmann, S., Baesens, B., Seowd, H., and Thomas, L. (2015). Benchmarking state-of-theart classification algorithms for credit scoring: An update of research. *The Journal of the Operational Research Society*, 247(1), 124-136. https://doi.org/10.1016/j.ejor.2015.05.030
- Lu, W. (2008). Maximum likelihood estimation in the proportional hazards cure model. *Ann Inst Stat Math*, 60, 545-574. https://doi.org/10.1007/s10463-007-0120-x
- Luenberger, D. and Ye, Y. (2008). *Linear and Nonlinear Programming, 3 ed.* Springer, New York, NY.
- Liu, H. and Shen, Y. (2009). A Semiparametric Regression Cure Model for Interval-Censored Data *Journal of the American Statistical Association*, 104(487), 1168-1178. https://doi.org/10.1198/jasa.2009.tm07494
- Ma, J. (2010). Positively constrained multiplicative iterative algorithm for maximum penalized likelihood tomographic reconstruction. *IEEE Transactions on Nuclear Science*, 57(1), 181-192. https://doi.org/10.1109/TNS.2009.2034462
- Ma, J., Heritier, S., and Lô, S. (2014). On the maximum penalized likelihood approach for proportional hazard models with right censored survival data. *Computational Statistics and Data Analysis*, 74, 142-156. https://doi.org/10.1016/j.csda.2014.01.005
- Murphy, S. (1994) Consistency in a proportional hazards model incorporating a random effect. *Annals of Statistics*, 22(2), 721–731. https://www.jstor.org/stable/2242287
- Murphy, S. (1995) Asymptotic theory for the frailty model. *Annals of Statistics*, 23(1), 182-198. https://www.jstor.org/stable/2242406
- Patilea, V., VanKeilegom, I. (2017). A General Approach for Cure Models in Survival Analysis. *arXiv: Statistics Theory*, 1-33. https://arxiv.org/pdf/1701.03769
- Peng, Y. and Dear, K. (2000). A Nonparametric Mixture Model for Cure Rate Estimation. *Biometrics*, 56(1), 237-243. https://www.jstor.org/stable/2677127
- Siddiqi, N. (2005). *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Wiley and SAS Business Series.
- Stepanova, M. and Thomas, L. (2002). Survival analysis methods for personal loan data. *Operations Research*, 50(2), 277-289. https://doi.org/10.1287/opre.50.2.277.426

- Sy, J. and Taylor, J. (2000). Estimation in a Cox Proportional Hazards Cure Model. *Biometrics*, 56(1), 227-236. https://doi.org/10.1111/j.0006-341X.2000.00227.x
- Thackham, M. and Ma, J. (2020). On maximum likelihood estimation of the semi-parametric Cox model with time-varying covariates. *Journal of Applied Statistics*, 47(9), 1511-1528. https://doi.org/10.1080/02664763.2019.1681946
- Thackham, M. and Ma, J. (2020). On maximum likelihood estimation of competing risks using the cause-specific semi-parametric Cox model with time-varying covariates An application to credit risk *Journal of the Operational Research Society*, xx(y), aaa-bbb. https://doi.org/10.1080/01605682.2020.1800418
- Tong, T., Mues, C., and Thomas, L. (2012). Mixture cure models in credit scoring: If and when borrowers default. *European Journal of Operational Research*, 218(1). 132-139. https://doi.org/10.1016/j.ejor.2011.10.007
- Wong, W. H. and Severini, T. A. (1991). On maximum likelihood estimation in infinite dimensional parameter spaces. Ann. Statist., 19(2), 603-632. https://doi.org/10.1214/aos/1176348113
- Wycinka, E. (2019) Competing Risk Models of Default in the Presence of Early Repayments. Econometrica, 23(2), 99–120. https://doi.org/10.15611/eada.2019.2.07
- Xu, J. and Scott-Long, J. (2005). Confidence intervals for predicted outcomes in regression models for categorical outcomes. *The Stata Journal*, 5(4), 537–559. https://doi.org/10.1177/1536867X0500500405
- Xu, J., Ma, J., Connors, M. H. and Brodaty, H. (2018). Proportional hazard model estimation under dependent censoring using copulas and penalized likelihood. *Statistics in Medicine*, 37(14), 2238-2251. https://doi.org/10.1002/sim.7651
- Yu, Y. and Ruppert, D. (2002). Penalized spline estimation for partially linear single index models. J. Amer. Statist. Assoc., 97(460), 1042-1054. https://doi.org/10.1198/016214502388618861
- Zangwill, W. (1969). *Nonlinear Programming: A Unified Approach*. Prentice-Hall, New Jersey.

6

Conclusion and Discussion

The core business of credit-granting institutions is lending money to customers, some of whom may default on their obligation to repay in a timely manner monies they contractually owe (namely principal, interest and fees). This exposes the institution to credit risk, which is "the potential loss a bank would suffer if a borrower fails to meets its obligations" (Apostolik et al., 2009). Lam (2003) details how risk quantification forms a vital component of effective risk management, in turn enabling application of appropriate risk controls so the institution's residual profile risk is maintained within risk appetite. Effective risk management by individual institutions have wider benefits to both the economy and society. In Australia, Authorised Deposit-Taking Institutions had total loans and advances of \$2.8 trillion (APRA, 2020), exceeding the country's entire 2019 Gross Domestic Product of \$1.9 trillion (ABS, 2020). As a concentrated financial market place, with many large and few small credit-granting institutions, a small disruption to the financial system from an institution that poorly manages its own risks can cause, as evidenced in the Financial Crisis of 2008, instability in other firms and ultimately economic recessions.

Quantitative credit risk management techniques estimate the frequency of customer default and loss severity in the event of default. Statistical estimates are formulated for metrics covering: Probability of Default (PD); Exposure at Default (EAD); and Loss Given Default (LGD). Together, PD, EAD and LGD are used as inputs to calculate expected loss (EL), which inform loan-loss provisions required under international accounting standards (IASB (2014), FASB (2016)), as well calculating unexpected loss (UL) required by institutions granted regulatory approval under the Basel Accords (BIS, 2006) to use the Advanced Internal Ratings Based (A-IRB) Approach for minimum credit capital. This thesis centred on improvements to methods quantifying the risk of credit default used for PD. Institutions already deploy analytical techniques using large amounts of their own internal data to quantify the risk of credit default and the refinements and improvements in this thesis could further assist institutions control their credit risk profile. See Thackham and Ma (2020) and Tong et al. (2012) respectively for example background on EAD and LGD.

To meet regulatory and audit approval, credit models need to have intuitive drivers and an understandable functional form. Regression techniques are therefore regularly used to estimate the risk of credit default and while logistic regression is common (Baesens et al. (2003), Lessmann et al. (2015)), several studies detail survival models achieve comparable accuracy to logistic regression but provide additional benefits (see fore example: Bellotti and Crook (2009), Stepanova and Thomas (2002) and Tong et al. (2012)). Chief benefits are inclusion of censored data and capability for model predictions across multiple time horizons, as logistic regression discards censored data and predicts over a single horizon. Other benefits of survival models include extensions for time-varying covariates, competing risks and mixture-cure approaches, which provide additional flexibility.

Survival analysis follows subjects in anticipation they encounter an event of interest. Such studies span not just finance but a wide array of disciplines, such as biomedical science and industrial life testing (Kalbfleisch and Prentice, 2002). Actuaries pioneered survival analysis (Fisher and Lin, 1999), with English astronomer Edmund Halley (1693) constructing the first life table of human mortality. Competing risk analysis (also known to Actuaries as "multiple decrements") was first used by the Swiss Physicist Daniel Bernoulli (1760) who extended Halley's (1693) life table method to demonstrate the increase in human survival if the competing risk of small pox were eliminated as a cause of death. In more contemporary

times, David Cox introduced the method of partial likelihood (1972, 1975), marking the start of regression analysis for time to event data, with the semi-parametric Cox model remaining a common and favoured regression technique for analysing such data (Ren and Zhou, 2011). This model has since been extended in numerous ways, with the more notable extensions explored in this thesis being those by Crowley and Hu (1977) to cater for time-varying covariates, and by (for example) Sy and Taylor (2000) to cater for mixture-cure models. This thesis contains via two published papers and one manuscript prepared in readiness for publication chapters on each of the Cox model with a single event of interest, competing risk analysis, and mixture-cure analysis, with a clear focus in the application of each of these approaches in the domain of credit risk modelling.

The first paper was published in the Journal of Applied Statistics (Thackham and Ma, 2020a) and concerns a computational approach to estimate the Cox model with time-varying covariates. Despite its many applications, the partial likelihood method used to estimate the Cox model with time-varying covariates contains two distinct shortcomings: (1) the baseline hazard is not directly estimated by the partial likelihood method, so that recovery of survival probabilities requires a further estimation step after fitting, such as that provided by Breslow (1972); and (2) the partial likelihood does not produce a covariance matrix for both fitted parameters and the baseline hazard, meaning joint inferences of the model parameters cannot be made. The paper developed a new methodology to address these two shortcomings. It does this by simultaneously estimating using maximum likelihood both regression coefficients and the baseline hazard whose sample design include uninformative right-censoring and time-varying covariates. Our approach adds to the literature by: (1) estimating model parameters using maximum likelihood; and (2) providing an estimate of the baseline hazard using a piece-wise constant basis which removes reliance on a secondary estimation step. We develop the necessary theory to estimate our model, including gradient vectors and the Hessian matrix, and implement this in the R programming language. Our approach extended the Newton Multiplicative-Iterative method of Ma et al. (2014) in order to jointly estimate the regression parameters and baseline hazard, which addresses the steep computational challenge of needing to respect the non-negativity constraint of the baseline hazard. We compare our proposed method with the partial likelihood method in combination with the Breslow (1972) baseline hazard estimator, using both a simulation study and a real-world application to model the risk of credit default for home loans. The results of the simulation study show superior performance of the maximum likelihood method over the partial likelihood method to recover the true population parameters in small to moderate sized samples. When applied to a sample of home loans (which has a large sample size), our results show that both baseline and time-varying estimated regression coefficients agree closely between the approaches, however the maximum likelihood estimate of the baseline hazard has markedly lower volatility.

The second paper was published in the Journal of Operations Research (Thackham and Ma, 2020b) concerned a computational approach to estimate the semi-parametric causespecific hazard Cox model with time-varying covariates of Prentice et al. (1978). It builds on Thackham and Ma (2020a) by extending it to an applied credit risk setting where there are two competing risks, these being the risk the customer defaults on their loan (which is the event of interest), and the competing risk that the customer fully repays all monies contractually owed. The paper outlines the full log-likelihood of the model with an arbitrary but finite number of competing risks, going on to devise an algorithm to estimate all the baseline regression coefficients, time-varying regression coefficients, and baseline hazards. We fit a competing risk model using our methodology for the risks of both loan default and loan repayment, demonstrating our method provides intuitive parameter estimates comparable to those returned using partial likelihood. Our method goes on to clearly confirm the shapes of the baseline hazards for the risks of loan default and loan repayment while also providing a 95% asymptotic confidence interval estimated via inverting the negative of the Hessian matrix readily produced by our maximum likelihood approach. We additionally show via simulation specifically devised to simulate competing risk data that our technique produces regression coefficient estimates with lower bias in small samples with heavy censoring.

The third paper was a manuscript prepared for submission and concerned a computational approach to estimate the semi-parametric mixture-cure model where the model assumes subjects are from two distinct populations; (1) those susceptible to default; and (2) those no longer susceptible and hence "cured". A common estimation approach for this model uses the Expectation Maximisation (EM) algorithm of Dempster et al. (1977). In this paper we apply a maximum likelihood approach to estimate the mixture-cure Cox model. Our approach starts with the full log-likelihood for which we calculate the gradient vector and Hessian

matrix and devise a computational scheme that extends further the approach of Thackham and Ma (2020a) and Thackham and Ma (2020b) using an amended version of the Newton Multiplicative-Iterative method of Ma et al. (2014). Using this constrained optimisation, we jointly estimate both regression parameters and the baseline hazard where the constraint ensures the hazard's non-negativity. This provides two key benefits over using the EM algorithm to estimate the model: (1) an asymptotic variance matrix for all model parameters is produced allowing joint inference on regression parameters and the baseline hazard without the need to bootstrap; and (2) a continuous baseline hazard estimate is returned that is directly estimated without (for example) a logarithm transform or an extra calculations.

There are a number of avenues of future research. The first avenue, the methodologies could be tested against other models, such as accelerated failure-time models, logistic regression models, and multinomial regression models. The second avenue, an understanding of how methodologies generalise to unseen data by using training and testing samples could be carried out. The third avenue, the mixture-cure methodology could be extended to include both time-varying covariates and/or competing risks.

In closing, our research in this thesis provides several enhancements the prevailing available quantitative credit risk management methodologies to estimate the risk of credit default using survival analysis, spanning contributions in survival analysis, competing risk analysis, and mixture cure analysis. Our approaches apply constrained maximum likelihood estimation, where the constraint ensures non-negative of baseline hazards, providing comparably accurate parameter estimates to prevailing approaches while also retuning less volatile baseline hazard estimates allowing clearer determination of their shapes. The significance of these baseline hazards can be easily discerned using the asymptotic variances returned by our maximum likelihood estimation approach.



Appendix - Implementation and Simulation

This appendix supports this thesis by providing additional background and the R code to implement our methodologies. This appendix consists of five sections:

- 1. Implementation of the Cox Model with time-varying covariates
- 2. Simulation of survival data with time-varying covariates
- 3. Simulation of competing risk data with time-varying covariates
- 4. Implementation of the mixture-cure model
- 5. Simulation of mixture-cure data

A.1 Model Implementation - Cox Model with TVC

In this section of the appendix, we discuss the model implementation of the Newton Multiplicative-Iterative Algorithm for constrained optimisation. This implementation is used for both Paper 1 which focussed on survival analysis, as well as paper 2 which focussed on competing risk analysis

A.1.1 Two Input data.frames

A key requirement of our implementation is the need for two input dataframes: one each for the baseline and time-varying covariates. The best manner in which to demonstrate this is via a small worked example. Suppose we have n = 3 subjects, for which have p = 2baseline and q = 2 time-varying covariates. Suppose further that for our data there are three event times at t = 3, 1 and 2 respectively for subjects 1, 2 and 3. The first of the two input dataframes governs baseline covariates, and contain one row for each subject. The columns detail the subject identifier, event time, and status (1 for observed event, 0 otherwise). The remaining p = 2 columns contain the data for the baseline covariates. This is demonstrated in table A.1 below.

Subject	Time	Status	x_1	<i>x</i> ₂
1	3	1	104	$\begin{array}{c}1\\0\\1\end{array}$
2	1	1	55	
3	2	1	23	

Table A.1: An Example Baseline Dataframe

The second of the two input dataframes governs the time-varying covariates, containing multiple rows for each subject (as outlined previously). Based on the unique observed event times from the data (in our case, these are the times t = 1, 2 and 3), each subject will have a record for each of these observed event times, up until when the given subject leaves the study. Table A.2 demonstrates this for our example. Subject i = 1 has an event time of $t_1 = 3$, so it has three records in the dataframe. Subject i = 2 has an event time of $t_2 = 1$, so it has only one record while subject i = 3 has an event time of $t_3 = 2$, so it has two records. The columns for this dataframe begin with a subject identifier, as well as a variable recording the j_i time-points (r_{ij_i}) when the time-varying covariates are measured for each of the *i* subjects. The final q = 2 columns contain the values of the time-varying covariates. Note that the key

feature of this dataframe is that the values of the time-varying covariates can potentially (but not necessarily) change over time for a given subject.



Table A.2: Example Time-Varying Covariate Data Frame

A key difficulty to overcome in the model implementation is how to undertake calculations for this expanded time-varying covariate dataframe. One natural method to do this in R is to separate the Z matrix into a list of *n* matrices, one each for every individual subject, and use functions such as lapply and mapply to undertake the necessary calculations. To demonstrate this for the data in tables A.1 and A.2, the baseline covariate matrix (X) and time-varying covariate matrix (Z), as defined (in the general case) earlier in the thesis are

$$X = \begin{bmatrix} x_{11} = 104 & x_{12} = 1\\ x_{21} = 55 & x_{22} = 0\\ x_{31} = 23 & x_{32} = 1 \end{bmatrix}$$
(A.1)

and

$$Z = \begin{bmatrix} z_1(r_{11}) = 5 & z_2(r_{11}) = 1\\ z_1(r_{12}) = 4 & z_2(r_{12}) = 0\\ z_1(r_{13}) = 2 & z_2(r_{13}) = 1\\ z_1(r_{21}) = 8 & z_2(r_{21}) = 0\\ z_1(r_{31}) = 3 & z_2(r_{31}) = 1\\ z_1(r_{32}) = 2 & z_2(r_{32}) = 1 \end{bmatrix}.$$
(A.2)

A.1.2 Fitting a Model and Available Controls

A.1.2.1 *coxph_mlt()*

As the user interface to our model implementation we create the coxph_mlt() function, which will jointly estimate regression coefficients and the baseline hazard of the Cox model using maximum likihood with time-varying covariates. The function is called in the following manner:

coxph_mlt (formula, data, formula.z, data.z, riji, subject, control, ...)

A call to the *coxph_mlt()* function requires the following six mandatory arguments, as well

as allowing for additional optional arguments which are passed to the *coxph_mlt.control()*:

formula	A formula object, with the response on the left of a ~ operator, and baseline covariates on the right separated by a "+" sign. The response must be a survival object as returned by the <i>survival::Surv()</i> function. A value of status=1 signifies that subject <i>i</i> was observed to have an event, while status=0 signifies the subject was censored. For example: <i>Surv(time , status)</i> ~ $x1 + x2$.
data	The baseline data.frame that contains the baseline covariate information, as well as the event time and status indicator used in the formula object. The dataframe also requires each entry to be signified by a unique subject identifier for each subject i .
formula.z	A linear predictor for the time-varying covariates. This should not be a formula object, but covariates need to be preceded by a ~ operator prior to listing the time-varying covariates on the right separated by a "+" sign. For example: <i>Surv(start, end , statusLong)</i> ~ $z1 + z2$.
data.z	The time-varying covariate data.frame, that has been expanded so that there is one record for every event time for every subject, up to the time that the subject leaves the study. For subject <i>i</i> , there are j_i records in the dataframe. One method to achieve this is to use the function <i>survival::survSplit()</i>
riji	The j_i time-points at which the time varying covariates are observed for subject <i>i</i> .
subject	The unique identifier for subject <i>i</i> .
	Other (optional) arguments which are passed to the control function $coxph_mlt.control()$.

The function returns an object of the class *coxph_mlt*, which is a list containing the results of the fitting algorithm. Example code to call the function is

 $fit.MPLt <- coxph_mlt(formula = Surv(time, status) \sim x1 + x2,$

data = baseline, $formula.z = Surv(start, end, statusLong) \sim z1 + z2,$ riji = end, subject = id, $data.z = time_varying).$

A.1.2.2 coxph_mlt.control()

The *coxph_mlt.control* function has the following optional arguments that allow the user to control various aspects of the model fit. Any arguments not supplied by the user will be assigned the indicated default values.

basis	The basis or the baseline hazard. Options are either <i>uniform</i> or <i>mspline</i>
max.iter	The maximum number of iterations for the Newton Multiplicative-Iterative algorithm. The default value is 10,000.
n.events_basis	(only available for <i>basis="uniform"</i>) The number of observed events to include in each basis of a uniform baseline hazard. With n.obs representing the number of observed events. The default is calculated using $round(3.5log(n.obs) - 7.5)$.
order	(only available for <i>basis="mspline"</i>) order of the msplines. The default is 3.
range.quant	(only available for <i>basis="mspline"</i>) The range for which quantiles are used to form internal knots. The default is $c(0.075, .9)$.
tol	The convergence tolerance value, which is the smallest change in the parameter estimates between iterations that when achieved indicates convergence has occurred. The default value is 1×10^{-6} .
min.theta	The size for the individual elements in the θ vector that are considered indistinguishable from zero. The default value is 1×10^{-10} .
kappa	The initial step-size in the line search. The default value is 0.6.

A.1.3 Exploring a Fitted Model

Six new functions have been written that accept as input the returned results of a fitted model with *coxph_mlt* to predict, extract, summarise and plot aspects related to a fitted model.

- 1. summary of a fitted model (summary.coxph_mlt)
- 2. returning model coefficients in a data.frame (tidy.coxph_mlt)
- 3. returning model coefficients in a vector (coef.coxph_mlt)
- 4. returning residuals of a fitted model (residuals.coxph_mlt)
- graphs of the baseline hazard, baseline cumulative hazard and baseline survival functions (plot.coxph_mlt)
- returning model predicted values for a subject potentially not included in the original model training (predict.coxph_mlt)

All of these functions accept an object of class *coxph_mlt* resulting from a model fit using summary.coxph_mlt.

A.1.3.1 summary.coxph_mlt()

This returns to the R console a summary of the fitted model including: fitted parameter estimates with associated p-values and significance tests; details regarding the number of

observed evens and total records; and detail on the convergence of the model such as the number of iterations and achieved value of the log-likelihood. The input is:

object An object of class *coxph_mlt* resulting from summary.coxph_mlt

detail (optional) determine the level of detail returned. Available values are: $detail="RegOnly" to display on the estimated regression coefficients of <math>\beta$ and γ); and $detail="All" which additionally displays estimated baseline parameters <math>\theta$. The default is "RegOnly"

A.1.3.2 *tidy.coxph_mlt()*

This returns a *data.frame* summary of the fitted model including: fitted parameter estimates with associated p-values and significance tests. The input is:

object An object of class *coxph_mlt* resulting from summary.coxph_mlt

A.1.3.3 coef.coxph_mlt()

This returns a *list* with three elements of the fitted model parameters. Each element contains the fitted parameters of β , γ and θ . The input is:

object An object of class *coxph_mlt* resulting from summary.coxph_mlt

A.1.3.4 residuals.coxph_mlt()

This returns a *data.frame* containing the Martingale and the Cox-Snell residuals of a fitted model. The input is:

object An object of class *coxph_mlt* resulting from summary.coxph_mlt

A.1.3.5 predict.coxph_mlt()

This returns a *list* of model predicted values for a subject potentially not included in the original model training.. The input is:

object	An object of class <i>coxph_mlt</i> resulting from summary.coxph_mlt
newdata.short	A <i>data.frame</i> for a single subject containing baseline covariates, event time and status
newdata.long	A <i>data.frame</i> for a single subject containing time-varying covariates, start time, end time and status
A.1.3.6 plot.coxph_mlt()

This returns an object of class *ggplot* of a graph of either the baseline hazard, baseline cumulative hazard or baseline survival functions for a model fitted using *coxph_mlt()*. The input is:

```
object An object of class coxph_mlt resulting from summary.coxph_mlt
type (optional) which type of graph is produced. Available levels are
"Hazard", "CumHazard" or "Survival. The default is "Hazard".
```

A.1.3.7 predict.coxph_mlt()

This returns a *list* of model predicted values for a subject potentially not included in the original model training. The input is:

object	An object of class <i>coxph_mlt</i> resulting from summary.coxph_mlt
newdata.short	A <i>data.frame</i> for a single subject containing baseline covariates, event time and status
newdata.long	A <i>data.frame</i> for a single subject containing time-varying covariates start time, end time and status

A.1.4 R Code

```
coxph_mlt=function(formula,data,formula.z,riji,subject,data.z,subset,
spare,na.action,control,...){
#_____
# get and organise information
# (same tests as in coxph(), thanks to the survival package)
mc = match.call(expand.dots = FALSE)
m = match(c("formula","data","subset","na.action"),names(mc),0)
mc.oria = mc
mc = mc[c(1,m)]
if (m[1]==0){stop("A formula argument is required")}
data.name = if(m[2]!=0){mc[m[2]][[1]]}else{"-"}
mc[[1]] = as.name("model.frame")
mc$formula = if(missing(data)) terms(formula)
else
                terms(formula, data=data)
mf = eval(mc.parent.frame())
if (any(is.na(mf))) stop("Missing observations in the model variables")
if (nrow(mf) ==0) stop("No (non-missing) observations")
mt = attr(mf."terms")
# extract response
    = model.extract(mf, "response")
type = attr(y, "type")
if(!inherits(y, "Surv")){stop("Response must be a survival object")}
if(type!="right"&&type!="counting"){
stop(paste("Cox model doesn't support \"", type, "\" survival data",sep = ""))
}
#=============
```

```
#==================
# indicator for survival or censoring
t_i
           = y[,1L]
observed = y[,2L]==1L
c_i = ifelse(observed==T,1,0)
          = length(t_i)
n
n.obs
           = sum(observed)
#==================
#==
# control arguments
extraArgs <- list(...)</pre>
if (length(extraArgs)) {
controlargs <- names(formals(coxph_mpl.control))</pre>
m <- pmatch(names(extraArgs), controlargs, nomatch=0L)</pre>
if (any(m==0L))
stop(gettextf("Argument(s) %s not matched", names(extraArgs)[m==0L]),
domain = NA, call. = FALSE)
if (missing(control)) control <- coxph_mpl.control(n.obs, ...)</pre>
#==================
# ties
t_i.obs = t_i[observed]
ties = duplicated(t_i.obs)
t_i.obs = t_i.obs[!ties]
n.obs = length(t_i.obs)
#======
#=====
# X matrix
#X
            = model.matrix(mt, mf, contrasts)
          = model.matrix(mt, mf)
Х
          = X[,!apply(X, 2, function(x) all(x==x[1])), drop=FALSE]
Х
          = ncol(X)
р
#=================
#===
      ______
# knot sequence and psi matrices
knots = knots_mpl(control, t_i.obs, range(t_i))
      = knots$m
m
# Basis functions
psi = basis_mpl(t_i,knots,control$basis,control$order,which=1)
PSI = basis_mpl(t_i,knots,control$basis,control$order,which=2)
        = penalty_mpl(control,knots)
#R
R=matrix(rep(0,m*m), nrow=m, ncol=m)
#====
#================
# Z variables
var.z=all.names(formula.z)
var.z=var.z[6:length(var.z)]
var.z=var.z[(var.z=='+')==FALSE]
var.z.status=all.names(formula.z[2])[4]
# Narrow the Z matrix
var.riji=mc.orig[[which(names(mc.orig)=='riji')]]
var.subj=mc.orig[[which(names(mc.orig)=='subject')]]
var.riji=as.character(var.riji)
var.subj=as.character(var.subj)
Z=data.z[,c(var.subj, var.riji, var.z)]
```

Subj=Z[,var.subj] noZ=FALSE q=ncol(Z) # Last value of Z matrix for each subject tmp=data.table(Subj=data.z[,var.subj],time=data.z[,var.riji], data.z[,var.z]) tmp2=tmp[, .SD[.N], by=Subj][,c('Subj','time'):=NULL] last.z=data.matrix(tmp2, rownames.force = NA) #==== # Times for all measures t_i_z=Z[,var.riji] # Basis functions Npsi = basis_mpl(t_i_z,knots,control\$basis,control\$order,which=1) NPSI = basis_mpl(t_i_z,knots,control\$basis,control\$order,which=2) NPSI = ifelse(NPSI<0.0.NPSI)</pre> #== #== # Difference in zPSI # First differences, but retaining the first element - Nxm NPSIdiff = data.frame(Subj=Z[,var.subj], NPSI, stringsAsFactors=FALSE) %>% group_by(Subj) %>% # Lagged difference, but not for first value mutate_at(vars(-group_cols()),list(~ifelse(row_number()==1,.,. - lag(.)))) %>% #mutate_all(funs(ifelse(row_number()==1,.,. - lag(.)))) %>% # Set NA to zero mutate_at(vars(-group_cols()),list(~ifelse(is.na(.),0,.))) %>% ungroup(Subj) %>% select(-Subj) %>% as.matrix # Initial value for Gamma q=length(var.z) Gamma=as.matrix(rep(0,q)) base::rownames(Gamma)=var.z # Most narrow version of Z Z=as.matrix(data.z[.c(var.z)]) longstatus=as.matrix(data.z[,c(var.z.status)]) #=== #========== # Fir the model lambda = control\$smooth = rep(0,p)Beta = rep(1,knots\$m) Theta correction = 1 full.iter = 0 this.max.iter=control\$max.iter[1] this.max.iter=1 for(iter in 1:this.max.iter){ fit <- coxphfit(</pre> status = as.integer(observed), longstatus=longstatus, X=X, Z=Z, last.z=last.z, Subj=Subj, R = R, psi = psi, PSI = PSI, Npsi=Npsi, NPSI=NPSI, NPSIdiff=NPSIdiff, Beta0 = Beta, Theta0 = Theta/correction, Gamma0=Gamma, lambda = as.double(lambda), kappa = control\$kappa, convVal = control\$tol, minTheta = control\$epsilon, maxiter = control\$max.iter[2]) 3 #-----#============

if(control\$max.iter[1]>1) control\$smooth = lambda	
<pre>M_theta_m1 = fit\$coef\$Theta</pre>	
<pre>H = as.matrix(fit\$matricies\$H) ;rownames(H)=colnames(H)=NULL</pre>	
<pre>p = length(fit\$coef\$Beta)</pre>	
<pre>m = length(fit\$coef\$Theta)</pre>	
<pre>q = length(fit\$coef\$Gamma)</pre>	
$Minv_2 = Hinv = matrix(0, p+q+m, p+q+m)$	
<pre>#pos = c(rep(IKUE,p), rep(IKUE,q), !(abs(M_theta_mI)<ie-5) !(fit\$gradiheta<="" &="" -ie-2))="" <="" pre=""></ie-5)></pre>	
$\mu_{\text{max}} = C(ep((koe, p), ep((koe, q), (abs(n_theta_mi) < ie^{-3} & (iitsorautheta < ie^{-2})))$	
π	
#======================================	
# Hessian	
<pre>temp = try(solve(H[pos,pos]),silent=TRUE)</pre>	
if(class(temp)!="try-error"){	
<pre>Hinv[pos,pos] = temp</pre>	
cov_NuNu_H = Hinv	
<pre>se.Eta_H = suppressWarnings(sqrt(diag(cov_NuNu_H)))</pre>	
}else{	
cov_NuNu_H = matrix(NA,p+m,p+m)	
se.Eta_H = rep(NA,p+m)	
}	
#=========	
#	
" # Gravh data	
# Grid to calcuate h0, h0.low, h0.high, H0, S0	
numPoints = 1000	
<pre>grid = seq(knots\$Alpha[1],max(knots\$Alpha),length=numPoints)</pre>	
# Evaultate the basis function psi(t) at the grid values	
M_psi_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=1)	
# Fuculture the basic function PFT(t) at the smid values	
# Evaultate the basis function PSI(t) at the grid values M PSI $x_m = basis mpl(grid knots basis control basis order control forder which=2)$	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2)</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, 50 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0)</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0)</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)]</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pma(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0)</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta be be we were the set of the s</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.high We have #DSI_Xm%*%Theta.high</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.low H0.high=M_DSI_Xm%*%Theta.low</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.high H0.low =M_PSI_Xm%*%Theta.high S0_low=exp(=W0)</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.high H0.low =M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.high S0.low=exp(-H0.low) 50 bisherm(U0 bish)</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.ligh H0.low =M_PSI_Xm%*%Theta.ligh S0.low=exp(-H0.low) S0.high=exp(-H0.high)</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, 50 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, 50 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)=1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)=1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Thet</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, 50 h0= M_psi_Xm%*%fit\$coef\$Theta H0= M_PSI_Xm%*%fit\$coef\$Theta S0=exp(-H0) # h0,H0, 50 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.low h0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.high S0.low=exp(-H0.low) S0.high=exp(-H0.high) # Save the graph data graphData=data.frame(t=grid, h0, h0.low, h0.high, H0. H0.low, H0.high, S0. 50.low. S0.high)</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=controlSbasis,order=controlSorder,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fitScoefSTheta H0= M_PSI_Xm%*%fitScoefSTheta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fitScoefSTheta)=1.96*se.Theta,0) Theta.high=as.numeric(fitScoefSTheta)=1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.high S0.low=exp(-H0.low) S0.high=exp(-H0.high) # Save the graph data graphData=data.frame(t=grid, h0, h0.low, h0.high, H0, H0.low, H0.high, S0, S0.low, S0.high) #====================================</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=controlSbasis,order=controlSorder,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fitScoefSTheta H0= M_PSI_Xm%*%fitScoefSTheta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fitScoefSTheta)-1.96*se.Theta,0) Theta.high=as.numeric(fitScoefSTheta)+1.96*se.Theta h0.low =M_psi_Xm%*%Theta.low h0.high=M_psi_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.high S0.low=exp(-H0.low) S0.high=exp(-H0.high) # Save the graph data graphData=data.frame(t=grid, h0, h0.low, h0.high, H0, H0.low, H0.high, S0, S0.low, S0.high) #====================================</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=controlSbasis,order=controlSorder,which=2) # h0,H0, S0 h0= M_pSi_Xm%*%fitScoefSTheta H0= M_PSI_Xm%*%fitScoefSTheta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fitScoefSTheta)-1.96*se.Theta,0) Theta.high=as.numeric(fitScoefSTheta)+1.96*se.Theta h0.low =M_psi_Xm%*Theta.low h0.high=M_psi_Xm%*Theta.low H0.high=M_pSI_Xm%*Theta.low H0.high=M_PSI_Xm%*Theta.low H0.high=M_PSI_Xm%*Theta.high S0.low=exp(-H0.low) S0.high=exp(-H0.high) # Save the graph data graphData=data.frame(t=grid, h0, h0.low, h0.high, H0, H0.low, H0.high, S0, S0.low, S0.high) #====================================</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=controlSbasis,order=controlSorder,which=2) # h0,H0, S0 h0= M_PSi_Xm%*%fitScoefSTheta H0= M_PSI_Xm%*%fitScoefSTheta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fitScoefSTheta)-1.96*se.Theta,0) Theta.high=as.numeric(fitScoefSTheta)+1.96*se.Theta h0.low =M_PSi_Xm%*%Theta.low h0.high=M_PSi_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Theta.low H0.high=M_PSI_Amk*%Thet</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=controlSbasis,order=controlSorder,which=2) # h0,H0, S0 h0= M_PSi_Xm%*%fitScoefSTheta H0= M_PSI_Xm%*%fitScoefSTheta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fitScoefSTheta)-1.96*se.Theta,0) Theta.high=as.numeric(fitScoefSTheta)+1.96*se.Theta h0.low =M_PSi_Xm%*%Theta.low h0.high=M_PSi_Xm%*%Theta.low H0.high=M_PSI_Xm%*%Thet</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=controlSbasis,order=controlSorder,which=2) # h0,H0, S0 h0= M_psi_Xm%*%fitScoefSTheta H0= M_PSI_Xm%*%fitScoefSTheta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fitScoefSTheta)-1.96*se.Theta,0) Theta.high=as.numeric(fitScoefSTheta)+1.96*se.Theta h0.low =M_psi_Xm%*WTheta.low h0.high=M_psi_Xm%*WTheta.low H0.high=M_PSI_Xm%*WTheta.low H0.high=M_PSI_Xm%*WTheta.high S0.low=exp(-H0.low) S0.high=exp(-H0.high) # Save the graph data graphData=data.frame(t=grid, h0, h0.low, h0.high, H0, H0.low, H0.high, S0, S0.low, S0.high) # # Residuals # Find values of Z at time t_i for each subject</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=controlSbasis,order=controlSorder,which=2) # h0,H0, S0 h6= M_pSi_Xm%*%fitScoefSTheta H0= M_PSI_Xm%*%fitScoefSTheta S0=exp(-H0) # h0,H0, S0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fitScoefSTheta)-1.96*se.Theta,0) Theta.high=as.numeric(fitScoefSTheta)+1.96*se.Theta h0.low =M_psi_Xm%*WTheta.low h0.high=M_psi_Xm%*WTheta.low H0.high=M_PSI_Xm%*WThet</pre>	
<pre># Evaultate the basis function PSI(t) at the grid values M_PSI_Xm = basis_mpl(grid,knots,basis=control\$basis,order=control\$order,which=2) # h0,H0, 50 h0= M_psi_Xm%%fit\$coef\$Theta H0= M_PSI_Xm%%fit\$coef\$Theta S0=exp(-H0) # h0,H0, \$0 low and high se.Theta=se.Eta_H[(p+q+1):(p+q+m)] Theta.low =pmax(as.numeric(fit\$coef\$Theta)-1.96*se.Theta,0) Theta.high=as.numeric(fit\$coef\$Theta)+1.96*se.Theta h0.low =m_psi_Xm%%Theta.low h0.high=M_psi_Xm%%Theta.low H0.high=M_pSI_Xm%%Theta.low H0.high=M_PSI_Xm%%Theta.low H0.high=M_PSI_Xm%%Theta.low H0.high=M_PSI_Xm%%Theta.low H0.high=M_PSI_Mm%%Theta.high 50.low=exp(-H0.high) # Save the graph data graphData=data.frame(t=grid, h0, h0.low, h0.high, H0, H0.low, H0.high, 50, 50.low, \$0.high) #============== # Residuals # Find values of Z at time t_i for each subject # These are the last values of the Z matrix var.subj2=quo(!! sym(var.subj))</pre>	

134

group_by(!! var.subj2)%>%

```
mutate(rn=row number().
max.rn=max(rn)) %>%
filter(rn==max.rn) %>%
ungroup
last.Z.out=as.matrix(last.Z[,c(var.subj, var.riji)])
last.Z =as.matrix(last.Z[,c(var.z)])
# Calucate the fitted values for cumulative hazard at time t_i
H_t_i=PSI %*% fit$coef$Theta * exp(X %*% fit$coef$Beta + last.Z %*% fit$coef$Gamma)
# Margingale residuals
Martingale.Res=data.frame(Martingale.Res=c_i-H_t_i)
# Cox-Snell residuals
CoxSnell.Res=data.frame(CoxSnell=H t i)
# Status
Status=data.frame(Status=c_i)
# Residuals
Residuals=data.frame(last.Z.out, Status, X, last.Z, Martingale.Res, CoxSnell.Res)
#===
if(control$basis=='uniform'){
# Graph data
se.Theta=se.Eta_H[(p+q+1):(p+q+m)]
se.Theta2=se.Theta[c(1,1:m)]
Theta2=fit$coef$Theta[c(1,1:m)]
graphData_uniform=data.frame(Alpha=knots$Alpha,
Theta=Theta2,
se.Theta=se.Theta2,
low=Theta2-1.96*se.Theta2,
high=Theta2+1.96*se.Theta2)
}
else {
graphData_uniform=NA;
#========
#====
# output
fit$knots = knots
fit$control = control
fit$call = match.call()
fit$dim = list(n = n, n.obs = sum(observed), n.ties = sum(ties), p = p, q = q, m = knots$m)
fit$data = list(time = t_i, observed = observed, X = X, Z = Z, name = data.name,
graphData=graphData, graphData_uniform=graphData_uniform, Residuals=Residuals)
fit$matricies=list(H=H, loglik=fit$loglik, cov_NuNu_H=cov_NuNu_H, se.Eta_H=se.Eta_H,
psi=fit$matricies$psi, PSI=fit$matricies$PSI, NPSIdiff=NPSIdiff, H0starMu=fit$matricies$H0starMu,
NPSIdiff_by_eZGamma_by_Theta_by_Mu=fit$matricies$NPSIdiff_by_eZGamma_by_Theta_by_Mu, History=fit$History)
class(fit) = "coxph_mlt"
fit
#========
3
#~~
#~~
coxphfit <- function(status, longstatus, X, Z, last.z, Subj, R, psi, PSI,</pre>
Npsi, NPSI, NPSIdiff, Beta0, Gamma0, Theta0, lambda, kappa, convVal, minTheta, maxiter){
p = ncol(X)
n = nrow(X)
m = ncol(R)
N = dim(Z)
1 = p+m
```

```
# Initialise
Theta0 = as.matrix(Theta0) # mx1
PsiTheta = PSI %*% Theta0 # nxm x mx1 = nx1
psiTheta = psi %*% Theta0 # nxm x mx1 = nx1
Mu = exp(X %*% Beta0)  # nxp x px1 = nx1
PsiThetaMu <- PsiTheta*Mu # element-wise nx1
psiThetaMu <- psiTheta*Mu # element-wise nx1</pre>
RTheta = R %*% Theta0
                          # mxm x mx1 = mx1
keep.names.Gamma=base::rownames(Gamma0)
# Number of measurements for each subject
count=data.frame(Subj=Subj, stringsAsFactors=FALSE) %>%
group_by(Subj) %>%
summarise(n=n())
#======
# Initialise
q=ncol(Z)
1 = p+m+q
Zu=exp(last.z%*% Gamma0) # nxq x qx1 = nx1
# Initialise H0star using Gamma0 and Theta0
calc0=calc(NPSIdiff=NPSIdiff, Z=Z, Subj=Subj, thisGamma=Gamma0, thisTheta=Theta0)
H0star=calc0$H0star
loglik = -H0star*Mu + status*log(psiTheta*Mu*Zu)
# nx1 + element-wise nx1 = nx1 (ith component is the ith obs contribution to loglik)
pen <- as.numeric(crossprod(Theta0,RTheta))  # 1xn x nx1 = 1x1 scalar</pre>
ploglik0 = (1-lambda)*sum(loglik) - lambda*pen # 1x1 scalar, the penalised likelihood
Gamma=Gamma0
Reta=Beta0
Theta=Theta0
#===
# Save values
this.maxiter=maxiter
ploglikMat=matrix(rep(0,(this.maxiter+1)*3), ncol=3, nrow=this.maxiter+1)
ploglikMat[1,]=ploglik0
BetaMat =matrix(rep(0,(this.maxiter+1)*p), nrow=this.maxiter+1, ncol=p)
GammaMat=matrix(rep(0,(this.maxiter+1)*q), nrow=this.maxiter+1, ncol=q)
ThetaMat=matrix(rep(0,(this.maxiter+1)*m), nrow=this.maxiter+1, ncol=m)
BetaMat [1,]=Beta0
GammaMat[1,]=Gamma0
ThetaMat[1,]=Theta0
#===
#===
# Update Beta
for(iter in 1:maxiter){
# Update beta
StatusMinH0starMu = status-H0star*Mu
GradBeta <- t(X) %*% StatusMinH0starMu
HessianBeta <- as.matrix(crossprod(X, Diagonal(n=n, x=H0star*Mu)) %*% X)</pre>
StepBeta <- solve(HessianBeta) %*% GradBeta
Beta <- Beta0 + StepBeta
##
Mu = exp(X %*% Beta)
loglik = -H0star*Mu + status*log(psiTheta*Mu*Zu)
ploglik = (1-lambda)*sum(loglik) - lambda*pen
# Adapt Newton step if needed
r=0
while(ploglik < ploglik0){</pre>
```

A.1 Model Implementation - Cox Model with TVC

r=r+1 StepBeta = StepBeta/kappa Beta <- Beta0 + StepBeta Mu = exp(X %*% Beta) loglik = -H0star*Mu + status*log(psiTheta*Mu*Zu) ploglik = (1-lambda)*sum(loglik) - lambda*pen #cat("Beta: ",r) if (r>500) break ploglik0=ploglik ploglikMat[1+iter,1]=ploglik0 #_____ # Update gamma # Create H0diff - Nxm x mx1 = Nx1 H0diff=NPSIdiff %*% Theta # Elements matrix B - Nx1 MuRep=rep(Mu, count\$n) Belement=H0diff * exp(Z%*%Gamma) * MuRep ## Newton longStatusMinBelementexpandMu = longstatus-Belement GradGamma <- crossprod(Z, longStatusMinBelementexpandMu)</pre> HessianGamma <- crossprod(Z, Diagonal(n=length(Belement), x=Belement)) %*% Z</pre> StepGamma <- solve(HessianGamma) %*% GradGamma</pre> Gamma <- Gamma0 + StepGamma Gamma=as.matrix(Gamma) # Update H0star using Gamma and Theta0 Zu=exp(last.z%*% Gamma) calc0=calc(NPSIdiff=NPSIdiff, Z=Z, Subj=Subj, thisGamma=Gamma, thisTheta=Theta) H0star=calc0\$H0star loglik = -H@star*Mu + status*log(psiTheta*Mu*Zu) # nx1 + element-wise nx1 = nx1 (ith component is the ith obs contribution to loglik) ploglik = sum(loglik) # 1x1 scalar, the penalised likelihood # Adapt Newton step if needed r=0 while(ploglik < ploglik0){</pre> r=r+1StepGamma = StepGamma/kappa Gamma <- Gamma0 + StepGamma Gamma=as.matrix(Gamma) Zu=exp(last.z%*% Gamma) calc0=calc(NPSIdiff=NPSIdiff, Z=Z, Subj=Subj, thisGamma=Gamma, thisTheta=Theta) H0star=calc0\$H0star loglik = -H@star*Mu + status*log(psiTheta*Mu*Zu) # nx1 + element-wise nx1 = nx1 (ith component is the ith obs contribution to loglik) ploglik = (1-lambda)*sum(loglik) - lambda*pen # 1x1 scalar, the penalised likelihood #cat("Gamma: ",r) if (r>50000) break ploglik0=ploglik ploglikMat[1+iter,2]=ploglik0 #===== # Update theta W <- psi/matrix(psi %*% Theta,nrow=nrow(psi),ncol=ncol(psi), byrow=F) #nXm /(element-wise) nXm matrix WTstatus <- t(W) %*% status #mXn x nx1, result is mx1

PSIstar=calc0\$PSIstar PSIstarMu <- crossprod(PSIstar, Mu)

```
GradTheta <- (1-lambda)*(WTstatus-PSIstarMu) - 2*lambda*(R %*% Theta)</pre>
sTheta <- Theta/((1-lambda)*PSIstarMu + ifelse(RTheta>0, 2*lambda*(R %*% Theta0), 0) + 0.3)
StepTheta <- GradTheta*sTheta
Theta <- as.matrix(Theta0) + StepTheta
Theta[which(Theta<minTheta)]=minTheta
calc0=calc(NPSIdiff=NPSIdiff, Z=Z, Subj=Subj, thisGamma=Gamma, thisTheta=Theta)
H0star=calc0$H0star # nX1
psiTheta = psi %*% Theta # nXm x mX1, result is nX1
RTheta = R %*% Theta # mXm x mX1, result is mX1
loglik = -H0star*Mu + status*log(psiTheta*Mu*Zu)
pen <- as.numeric(crossprod(Theta,RTheta))</pre>
ploglik = (1-lambda)*sum(loglik) - lambda*pen
# Adapt Newton step if needed
r=0
while(ploglik < ploglik0){</pre>
r=r+1
StepTheta <- StepTheta/kappa
Theta <- as.matrix(Theta0) + StepTheta
Theta[which(Theta<minTheta)]=minTheta
calc0=calc(NPSIdiff=NPSIdiff, Z=Z, Subj=Subj, thisGamma=Gamma, thisTheta=Theta)
H0star=calc0$H0star
psiTheta = psi %*% Theta
psiThetaMu <- psiTheta*Mu
RTheta = R %*% Theta
loglik = -H0star*Mu + status*log(psiThetaMu)
pen <- as.numeric(crossprod(Theta,RTheta))</pre>
ploglik = (1-lambda)*sum(loglik) - lambda*pen
#cat("Theta: ",r)
if (r>500) break
}
# Save the penalised likelihood
ploglik0 <- ploglik
ploglikMat[1+iter,3]=ploglik0
# Check for convergence
varepsilon <- max(c(abs(Beta-Beta0),abs(Gamma-Gamma0),abs(Theta-Theta0)))</pre>
if (varepsilon<convVal) break
Beta0 <- Beta
Gamma0 <- Gamma
Theta0 <- Theta
BetaMat [1+iter,]=Beta0
GammaMat[1+iter,]=Gamma0
ThetaMat[1+iter,]=Theta0
#print(iter)
if ( (round(iter/10,6)-floor(iter/10)) == 0){
cat(iter, 'iterations...\n')
}
}
# =================
# ==
#Inference for Beta
H0starMu=H0star*Mu
V1 <- - t(X) %*% Diagonal(n=length(H0starMu), x=H0starMu) %*% X
colnames(V1)=rownames(V1)=NULL
HessianBeta <- V1
# =====
# =
# Inference for Gamma
```

```
138
```

```
# Create H0diff - Nxm x mx1 = Nx1
H0diff=NPSIdiff %*% Theta
# Elements matrix B - Nx1
MuRep=rep(Mu. count$n)
Belement=H0diff * exp(Z%*%Gamma) * MuRep
V2= - t(Z) %*% Diagonal(n=length(Belement), x=Belement) %*% as.matrix(Z)
V2=as.matrix(V2)
HessianGamma <- V2
# ====
# ====
# Inference for Theta
V3 <- as.matrix(t(psi) %*% Diagonal(n=n, x=as.numeric(status/psiTheta^2)) %*% psi) #(mxn)x(nxn)x(nxm)=mxm #B
HessianTheta <- V3
# ====
# _____
# Inference for d21/dbeta dtheta
V13 = - t(X) %*% Diagonal(n=length(Mu), x=Mu) %*% calc0$PSIstar
V13=as.matrix(V13)
colnames(V13)=rownames(V13)=NULL
# ====
# Inference for dl/dbeta dgamma
# Replicate the rows of X j_{i} times each - result is Nxp matrix
timesToRep=count$n
rows <- rep( 1:nrow(X) , timesToRep)</pre>
Xrep=matrix(X[rows,], ncol=ncol(X))
V12 = - t(Xrep) %*% Diagonal(n=length(Belement), x=Belement) %*% Z
V12=as.matrix(V12)
# ==
# =================
# Inference for dl/dgamma dtheta
# Replicate the rows of Mu j_{\{i\}} times each - result is Nx1 vector
timesToRep=count$n
rows <- rep( 1:nrow(Mu) , timesToRep)</pre>
Murep=matrix(Mu[rows,], ncol=1)
V23= - t(Z) %*% Diagonal(n=length(Murep), x=Murep) %*% calc0$NPSIdiff_by_eZGammaRep
V23=as.matrix(V23)
# =================
# =
# H matrix
row1=cbind( V1 , V12 , V13)
row2=cbind(t(V12), V2 , V23)
row3=cbind(t(V13), t(V23), -V3)
Halt=-1*rbind(row1, row2, row3)
H=Halt
M2=Halt
# =========
# =====
# Make sure Gamma has the right names;
base::rownames(Gamma)=keep.names.Gamma
# _____
# _____
#Output
return(list(coef=list(Beta=Beta, Gamma=Gamma, Theta=Theta),loglik=list(iter=iter, ploglik=ploglik[1], correction=ploglik[2],
```

ploglikMat=ploglikMat),matricies=list(H=H, H0starMu=H0starMu, psi=psi, PSI=PSI, H0star=H0star,

NPSIdiff_by_eZGammaRep=calc0\$NPSIdiff_by_eZGammaRep),GradTheta=GradTheta, History=list(BetaMat=BetaMat, GammaMat=GammaMat, ThetaMat=ThetaMat))) # == 3 #~, # Function to update H0star and PSIstar using the latest Gamma and Theta calc<-function(NPSIdiff, Z, Subj, thisGamma, thisTheta)</pre> # exp(zT.Gamma) - Nx1 eZGamma=exp(Z%*%thisGamma) # Replicate the number of columns for eZGamma - Nxm thism=dim(thisTheta)[1] thismRep=matrix(rep(1,thism), nrow=1) eZGammaRep = eZGamma %*% thismRep # Element wise multiplication - Nxm NPSIdiff_by_eZGammaRep=NPSIdiff * eZGammaRep # Summarise to create PSIstar - nXm #PSIstar.old = data.frame(Subj=Subj, NPSIdiff_by_eZGammaRep, stringsAsFactors=FALSE) %>% # group_by(Subj) %>% # # Sum the columns # summarise_all(sum) %>% # ungroup %>% # select(-Subj) %>% # as.matrix # Summarise to create PSIstar - nXm temp = data.table(Subj=Subj, NPSIdiff_by_eZGammaRep)[,j= lapply(.SD, sum), by=Subj][,Subj:=NULL] PSIstar=data.matrix(temp, rownames.force = NA) rm(temp) # Create H0star - nX1 H0star=PSIstar %*% thisTheta return(list(H0star=H0star,PSIstar=PSIstar, NPSIdiff_by_eZGammaRep=NPSIdiff_by_eZGammaRep)) } #~ #~ # function to predict: # h0, H0, S0 for a baseline subject and # h. H and S for a non-basline subject predict.coxph_mlt <- function(object, newdata.short, newdata.long) {</pre> #-----# Regression coefficients valuesBeta =data.frame(xVars=rownames(object\$coef[1:2]\$Beta), BetaCoef=object\$coef[1:2]\$Beta, stringsAsFactors=F) rownames(valuesBeta)=NULL $values {\tt Gamma} = {\tt data.frame(zVars=rownames(object$coef[1:2]${\tt Gamma}, {\tt Gamma}, {\tt Gamma}, {\tt Gamma}, {\tt strings} {\tt AsFactors=F}) \\$ rownames(valuesGamma)=NULL #-----#-----# Time the.time.var=as.character(object\$call\$riji) #-----#----- $\ensuremath{\texttt{\#}}$ Create X and beta vectors

A.1 Model Implementation - Cox Model with TVC

```
# Need to cater for dummies and continuous variables
numsTest = t(t(unlist(lapply(newdata.short, is.numeric))))
numsTest2 = data.frame(V1=rownames(numsTest), V3=numsTest[,1], stringsAsFactors=F)
# Transpose input data
temp=t(newdata.short)
newdata.short2 = data.frame(V1=rownames(temp), V2=temp[,1], stringsAsFactors=F) %>%
left_join(numsTest2, by='V1') %>%
mutate(xVars = ifelse(V3==TRUE,paste0(V1), paste0(V1, V2))) %>%
mutate(Value = as.numeric(ifelse(V3==TRUE,V2, 1))) %>%
select(xVars, Value)
# Merge on the parameters
valuesBeta2 = valuesBeta %>%
left_join(newdata.short2, by='xVars') %>%
mutate(Value=ifelse(is.na(Value),0, Value))
# X and beta vectors
X=t(valuesBeta2[.3])
beta=t(t(valuesBeta2[,2]))
#-----
#----
# Create Z and gamma vectors
Z = newdata.long %>%
select(valuesGamma$zVars) %>%
as.matrix
gamma=t(t(valuesGamma[,2]))
#-----
#-----
# Create theta vectors
theta=object$coef$Theta
#-----
#-----
# Time
the.time = newdata.long %>%
ungroup %>%
select (.dots=the.time.var)
colnames(the.time)='time'
# Grid
#numPoints = max(object$knots$Alpha)-object$knots$Alpha[1]+1
#temp_grid = seq(object$knots$Alpha[1],max(object$knots$Alpha),length=numPoints)
#grid=sort(unique(c(the.time$time, temp_grid)))
numPoints = 4
temp_grid = seq(object$knots$Alpha[1],max(object$knots$Alpha),length=numPoints)
grid=sort(unique(c(temp_grid)))
# Evaultate the basis function psi(t) at the grid values
M_psi_Xm = basis_mpl(grid,object$knots,basis=object$control$basis,order=object$control$order,which=1)
# Evaultate the basis function PSI(t) at the grid values
M_PSI_Xm = basis_mpl(grid,object$knots,basis=object$control$basis,order=object$control$order,which=2)
# h0,H0, S0
h0= M_psi_Xm%*%object$coef$Theta
H0= M_PSI_Xm%*%object$coef$Theta
S0=exp(-H0)
# h0,H0, S0 low and high
p=object$dim$p
q=object$dim$q
```

m=object\$dim\$m

se.Theta=object\$matricies\$se.Eta_H[(p+q+1):(p+q+m)]
Theta.low =pmax(as.numeric(object\$coef\$Theta)-1.96*se.Theta,0)
Theta.high=as.numeric(object\$coef\$Theta)+1.96*se.Theta
h0.low =M_psi_Xm%*%matrix(Theta.low)
h0.high=M_psi_Xm%*%matrix(Theta.high)
H0.low =M_PSI_Xm%*%Theta.low
H0.high=M_PSI_Xm%*%Theta.high
S0.low=exp(-H0.low)
S0.high=exp(-H0.high)

Collate the baseline values baseline=data.frame(time=grid, h0=h0, h0.low=h0.low, h0.high=h0.high, H0=H0, H0.low=H0.low, H0.high=H0.high, S0=S0, S0.low=S0.low, S0.high=S0.high)

Z Linear predictor linPredZ=data.frame(time=the.time\$time, linPredZ=Z %*% gamma, Z)

X Linear predictor linPredX=X %*% beta

Linear predictor linPred=cbind(linPredZ, linPredX) %>% mutate(linPred=linPredZ + linPredX) %>% select(-linPredZ, -linPredX)

alltime #alltime=data.frame(time=unique(sort(c(grid, the.time\$time)))) alltime=data.frame(time=unique(sort(c(grid))))

Final output

Join the data by time alltime2 = alltime %>% left_join(baseline, by='time') %>% left_join(linPred, by='time')

Fill down intermedate values
alltime3 = alltime2 %>%
fill(h0,.direction='down') %>%
fill(h0.low,.direction='down') %>%
fill(h0.high,.direction='down') %>%
fill(H0,.direction='down') %>%
fill(S0,.direction='down') %>%
fill(linPred, .direction='down') %>%
fill(valuesGamma\$zVars, .direction='down')

Calculate the non-baseline values alltime4 = alltime3 %>% mutate(adj=exp(linPred), h=h0 * adj, H0.comp=h0*(time-lag(time)), H0.comp=ifelse(is.na(H0.comp),0,H0.comp), H0=cumsum(H0.comp), H=H0 * adj, S0=exp(-H0), S=exp(-H0), %>% select(-adj) #------

#----# Non-baseline subject

Calculate NPSIdiff

A.1 Model Implementation - Cox Model with TVC

NPSIdiff = data.frame(Subj=1, M_PSI_Xm, stringsAsFactors=FALSE) %>%
group_by(Subj) %>%
Lagged difference, but not for first value
mutate_at(vars(-group_cols()),list(~ifelse(row_number()==1,.,. - lag(.)))) %>%
Set NA to zero
mutate_at(vars(-group_cols()),list(~ifelse(is.na(.),0,.))) %>%
ungroup(Subj) %>%
select(-Subj) %>%
as.matrix

Z matrix, evaluated at the grid Zgrid = alltime4 %>% select(valuesGamma\$zVars) %>% as.matrix

Call calc0
calc0=calc(NPSIdiff=M_PSI_Xm, Z=Zgrid, Subj=1, thisGamma=gamma, thisTheta=theta)
#------

#-----

partial H / partial beta
Mu = linPredX
H0starMu=calc0\$H0star*Mu
dH_dBeta = t(X) %*% H0starMu
#------

#-----

partial H / partial gamma
Create H0diff - Nxm x mx1 = Nx1
H0diff=NPSIdiff %*% theta

Elements matrix B - Nx1
MuRep=rep(Mu, length(H0diff))
Belement=H0diff * exp(Zgrid %*% gamma) * MuRep

dH_dGamma = t(Zgrid) %*% Belement
#-----

#-----# partial H / partial theta
PSIstarMu = crossprod(calc0\$PSIstar, Mu)
dH_dTheta = PSIstarMu
#------

#-----dH_dEta=rbind(dH_dBeta, dH_dGamma, dH_dTheta)
#------

return(alltime4)

} #~

#~

summary.coxph_mlt <-function(object=NULL, detail='RegOnly'){
Dimensions
p=object\$dim\$p
q=object\$dim\$p
m=object\$dim\$m</pre>

Regression coefficents Coeficient=round(rbind(object\$coef\$Beta, object\$coef\$Gamma),8) Variable=t(t(row.names(Coeficient)))

Regression significance
se=round(object\$matricies\$se.Eta_H[1:(p+q)],8)

z=round(Coeficient/se,8) pval=round(pnorm(-abs(Coeficient/se), 0, 1),8)*2 # Compile table for regression estaimtes reg=data.frame(Variable=Variable, Coeficient=Coeficient, Std.Err=se, z=z, pval=pval, row.names=NULL, stringsAsFactors =FALSE) %>% mutate(sig=ifelse(pval <0.001, "***",</pre> ifelse(pval <0.01, "**", ifelse(pval <0.05, "*", ifelse(pval <0.1, ".",''))))) # h0(t) Basis Function coefficents Coeficient.Theta=object\$coef\$Theta Variable.Theta=paste0('Theta', seq(1,m,1)) # h0(t) Basis Function significance $\texttt{se.Theta=round(object$matricies$se.Eta_H[(p+q+1):(p+q+m)],20)}$ z.Theta=round(Coeficient.Theta/se.Theta,20) pval.Theta=round(pnorm(-abs(Coeficient.Theta/se.Theta), 0, 1),20)*2 # Compile table for h0(t) Basis Function estaimtes h0r=data.frame(Variable=Variable.Theta, Coeficient=Coeficient.Theta, Std.Err=se.Theta, z=z.Theta, pval=pval.Theta, row.names=NULL, stringsAsFactors =FALSE) %>% mutate(sig=ifelse(pval <0.001, "***",</pre> ifelse(pval <0.01, "**", ifelse(pval <0.05, "*", ifelse(pval <0.1, ".",''))))) # Significance details sigs="Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1" # Print the output cat('-----')
cat('-----') cat(' n n')cat('Details of the Model Call:') cat('\n') print(object\$call) cat('\n') cat('Model Details:') cat("\n Number of Subjects : ",object\$dim\$n) cat("\n Number of Events : ",object\$dim\$n.obs) cat("\n Number of Iterations : ",object\$matricies\$loglik\$iter) cat("\n Log-Likelihood : ",object\$matricies\$loglik\$ploglik) cat("\n h0(t) Basis Function : ",object\$control\$basis) cat("\n Number of Parameters : ",p,' baseline ,', q, 'time-varying, ', m, 'h0(t) basis') cat(' n n')cat('Parameter Estimates:') cat('\n') print(reg) if(detail=='All'){ cat('\n') cat('h0(t) Basis Function Parameter Estimates:') cat('\n') print(h0r) }

```
cat(sigs)
cat(' n n')
}
#~
#~-
tidy.coxph_mlt <-function(object=NULL){</pre>
# Dimensions
p=object$dim$p
q=object$dim$q
m=object$dim$m
# Regression coefficents
Coeficient=round(rbind(object$coef$Beta, object$coef$Gamma),8)
Variable=t(t(row.names(Coeficient)))
# Regression significance
se=round(object$matricies$se.Eta_H[1:(p+q)],8)
z=round(Coeficient/se.8)
pval=round(pnorm(-abs(Coeficient/se), 0, 1),8)*2
# Compile table for regression estaimtes
reg=data.frame(Variable=Variable,
Coeficient=Coeficient,
Std.Err=se,
z=z,
pval=pval,
row.names=NULL,
stringsAsFactors =FALSE) %>%
mutate(sig=ifelse(pval <0.001, "***",</pre>
ifelse(pval <0.01, "**",
ifelse(pval <0.05, "*",
ifelse(pval <0.1, ".",'')))))
# h0(t) Basis Function coefficents
Coeficient.Theta=object$coef$Theta
Variable.Theta=paste0('Theta', seq(1,m,1))
# h0(t) Basis Function significance
\texttt{se.Theta=round(object\$matricies\$se.Eta\_H[(p+q+1):(p+q+m)],8)}
z.Theta=round(Coeficient.Theta/se.Theta.8)
pval.Theta=round(pnorm(-abs(Coeficient.Theta/se.Theta), 0, 1),8)*2
# Compile table for h0(t) Basis Function estaimtes
h0r=data.frame(Variable=Variable.Theta,
Coeficient=Coeficient.Theta,
Std.Err=se.Theta,
z=z.Theta.
pval=pval.Theta,
row.names=NULL,
stringsAsFactors =FALSE) %>%
mutate(sig=ifelse(pval <0.001, "***",</pre>
ifelse(pval <0.01, "**",
ifelse(pval <0.05, "*",
ifelse(pval <0.1, ".",'')))))
out=list(coef=reg, h0.coef=h0r)
return(out)
}
#~~
                                                     coef.coxph_mlt=function(object) {
return(object$coef)
```

3 plot.coxph_mlt=function(object, type='Hazard') { if (type == 'Hazard'){ # Baseline hazard p1=ggplot(data=object\$data\$graphData) + geom_step(aes(x=t, y=h0, color='h0(t)')) + geom_step(aes(x=t, y=h0.low, color='95%CI')) + geom_step(aes(x=t, y=h0.high,color='95%CI')) + ggtitle('Baseline Hazard with 95% CI') + ylab('h0(t)') + theme(plot.title = element_text(hjust = 0.5), legend.position='bottom') + scale color discrete(name = "Series") } if (type == 'CumHazard'){ # Baseline cumulative hazard p1=ggplot(data=object\$data\$graphData) + geom_step(aes(x=t, y=H0, color='H0(t)')) + geom_step(aes(x=t, y=H0.low, color='95%CI')) + geom_step(aes(x=t, y=H0.high,color='95%CI')) + ggtitle('Cumulative Baseline Hazard with 95% CI') + ylab('H0(t)') + theme(plot.title = element_text(hjust = 0.5), legend.position='bottom') + scale_color_discrete(name = "Series") } if (type == 'Survival'){ # Baseline survival p1=ggplot(data=object\$data\$graphData) + geom_step(aes(x=t, y=S0, color='S0(t)')) + geom_step(aes(x=t, y=S0.low, color='95%CI')) + geom_step(aes(x=t, y=S0.high,color='95%CI')) + ggtitle('Baseline Survival with 95% CI') + ylab('SO(t)') + ylim(0,1)+ theme(plot.title = element_text(hjust = 0.5), legend.position='bottom') + scale color discrete(name = "Series") } return(p1) #~~ #~ residuals.coxph mlt=function(object) { return(object\$data\$Residuals) coxph_mpl.control <- function(n.obs=NULL, basis = "uniform", smooth = NULL, max.iter=c(50,1e+05), tol=1e-7,</pre> n.knots = NULL, n.events_basis = NULL, range.quant = c(0.075,.9), cover.sigma.quant = .25, cover.sigma.fixed=.25, min.theta = 1e-10, penalty = 2L, order = 3L, kappa = 1/.6, epsilon = 1e-50, ties = "unique", seed = NULL, knots.manual=NULL){ basis = basis.name_mpl(basis) max.iter = c(ifelse(is.null(smooth),ifelse(max.iter[1]>0,as.integer(max.iter[1]),50),1L), ifelse(max.iter[2]>0,as.integer(max.iter[2]),1e+05)) tol = ifelse(tol>0 & tol<1,tol,1e-7) = ifelse(order>0 & order<6.as.integer(order).3L)</pre> order min.theta = ifelse(min.theta>0 & min.theta<1e-3,min.theta,1e-10) = penalty.order_mpl(penalty,basis,order) penalty kappa = ifelse(kappa>1, kappa, 1/.6) cover.sigma.quant = ifelse(cover.sigma.quant>0 & cover.sigma.quant<0.4,cover.sigma.quant,0.3)</pre> cover.sigma.fixed = ifelse(cover.sigma.fixed>0 & cover.sigma.fixed<0.4,cover.sigma.fixed,0.3)</pre> if(all(range.quant<1) & all(range.quant>0) & length(range.quant)==2){

A.1 Model Implementation - Cox Model with TVC

```
range.guant = range.guant[order(range.guant)]
else{range.quant = c(0.075,.9)
if(is.null(n.knots)|sum(n.knots)<3|length(n.knots)!=2){
n.knots = if(basis!='uniform' & basis!='msplines'){c(0,20)}else{c(8,2)}
if(!is.null(n.events_basis)){
n.events_basis = ifelse(n.events_basis<1|n.events_basis>floor(n.obs/2),
max(round(3.5*log(n.obs)-7.5),1L),round(n.events_basis))
}else{n.events_basis = max(round(3.5*log(n.obs)-7.5),1L)}
if(!is.null(smooth)){
smooth = ifelse(smooth<0,0,smooth)</pre>
}else{smooth=0}
out = list(basis = basis, smooth = smooth, max.iter = max.iter, tol = tol,
order = order, penalty = penalty, n.knots = n.knots, range.quant = range.quant,
cover.sigma.quant = cover.sigma.quant, cover.sigma.fixed = cover.sigma.fixed,
n.events_basis = as.integer(n.events_basis), min.theta = min.theta, ties = ties,
seed = as.integer(seed), kappa = kappa, epsilon = epsilon,
knots.manual=knots.manual)
class(out) = "coxph_mpl.control"
out
}
#~~
basis.name_mpl <- function(k){</pre>
if(k == "discr"| k == "discretized" | k == "discretised" | k == "unif" | k == "uniform"){"uniform"
}else{if(k == "m" | k == "msplines" | k == "mspline"){"msplines"
}else{if(k == "gauss" | k == "gaussian"){"gaussian"
}else{if(k == "epa" | k == "epanechikov"){"epanechikov"
}else{stop("Unkown basis choice", call. = FALSE)}}}
}#~~
penalty.order_mpl <- function(p,basis,order){</pre>
p = as.integer(p)
switch(basis,
'uniform' = ifelse(p>0 & p<3,p,2),
'gaussian' = ifelse(p>0 & p<3,p,2),
'msplines' = order-1,
'epa'
         = 2)
3
#~
knots_mpl=function(control,events,range){
n.events = length(events)
## uniform
if(control$basis=="uniform"){
        = floor(n.events/control$n.events_basis)
m
n.i_u = round(control$n.events_basis*c(rep(1L,m-1L),1L+n.events/control$n.events_basis-m))
Alpha
         = c(range[1],unlist(lapply(split(events[order(events)],rep(1L:m,n.i_u)),max)))
if(Alpha[length(Alpha)]<range[2]){Alpha[length(Alpha)]=range[2]}
if(!(is.null(control$knots.manual))){Alpha=c(control$knots.manual, range)}
Alpha = Alpha[order(Alpha)]
Alpha=unique(sort(Alpha))
Alpha = unique(Alpha)
m
         = length(Alpha)-1
Delta
          = Alpha[2L:(m+1L)]-Alpha[1L:m]
list(m=m,Alpha=Alpha,Delta=Delta)
## other
}else{
# quantile knots
if(control$n.knots[1]>0){
```

```
Alpha1 = quantile(events,prob=seq(control$range.quant[1],control$range.quant[2],length=control$n.knots[1]))
}else{Alpha1 = NULL}
#control$n.knots[2] = max(control$n.knots[2]+2,2)
Alpha2 = c(range[1],seq(ifelse(control$n.knots[1]==0,range[1],max(Alpha1)),
range[2],length=control$n.knots[2])[-1])
if(control$n.knots[2]==0){Alpha2=range}
Alpha = unique(c(Alpha1,Alpha2))
Alpha = Alpha[order(Alpha)]
Alpha=unique(sort(Alpha))
n.Alpha = length(Alpha)
## gaussian-basis
if(control$basis=="gaussian"){
Sigma = Delta = rep(0,n.Alpha)
for(aw in 1:n.Alpha){
if(aw>1 & aw<(n.Alpha-control$n.knots[2]+2)){
while(sum(events>(Alpha[aw]-2*Sigma[aw])&events<(Alpha[aw]+2*Sigma[aw]))<(n.events*control$cover.sigma.quant)){</pre>
Sigma[aw] = Sigma[aw] + 0.001}
}else{Sigma[aw] = control$cover.sigma.fixed*(Alpha[n.Alpha]-Alpha[1])/3}
Delta[aw]= pnorm((range[2]-Alpha[aw])/Sigma[aw])-
pnorm((range[1]-control$epsilon-Alpha[aw])/Sigma[aw])
list(m=n.Alpha, Alpha=Alpha, Sigma=Sigma, Delta=Delta)
## m-splines and epanechikov
}else{
m = n.Alpha+control$order-2
list(m=m, Alpha=Alpha, Delta=rep(1,m))
}
}
#,
basis_mpl = function(x,knots,basis,order,which=c(1,2)){
which.matrix = rep(T,2)
which.matrix[-which]=FALSE
n
         = length(x)
Alpha = knots$Alpha
Delta = knots$Delta
n.Alpha = length(Alpha)
        = ifelse(basis=="msplines"|basis=="epanechikov",n.Alpha+order-2,knots$m)
m
M_Psi_nm = M_psi_nm = matrix(0,n,m)
##
if(basis=="uniform"){
u_i = sapply(x,function(y,lim=Alpha[-1L])sum(lim<y)+1L)</pre>
for(i in 1:n){
M_psi_nm[i,u_i[i]] = 1
M_Psi_nm[i,1:u_i[i]] = c(if(u_i[i]>1){Delta[1:(u_i[i]-1)]},
x[i]-Alpha[u_i[i]])
}
##
}else{
if(basis=="gaussian"){
Sigma = knots$Sigma
for(u in 1:m){
M_psi_nm[,u] = dnorm((x-Alpha[u])/Sigma[u])/(Sigma[u]*Delta[u])
M_Psi_nm[,u] = (pnorm((x-Alpha[u])/Sigma[u])-
pnorm((Alpha[1]-Alpha[u])/Sigma[u]))/Delta[u]
##
}else{
seq1n = 1:n
Alpha_star = as.numeric(c(rep(Alpha[1],order-1L),Alpha,rep(Alpha[n.Alpha],order-1L)))
```

```
M psi nm
            = M Psi nm = cbind(M psi nm.0)
if(which.matrix[1]){
Alpha_star_x = sapply(x,function(y,lim=Alpha[-1L])sum(lim<y)+1L)+order-1L</pre>
if(basis=="msplines"){
M_psi_nm[(Alpha_star_x-1L)*n+seq1n]=1/(Alpha_star[Alpha_star_x+1]-Alpha_star[Alpha_star_x])
if(order>1){
for(ow in 2L:order){
uw_x = Alpha_star_x-ow+1L
for(pw in 0:(ow-1L)){
pos_x = (uw_x+pw-1L)*n+seq1n
\label{eq:m_psi_nm[pos_x]=(ow/((ow-1)*(\mbox{Alpha\_star[1:m+ow]-Alpha\_star[1:m]})))[uw\_x+pw]*
((x-Alpha_star[uw_x+pw])*M_psi_nm[pos_x]+
(Alpha_star[uw_x+pw+ow]-x)*M_psi_nm[pos_x+n])
3
}
}
# Epanechikov
}else{
uw x = Alpha star x-order+1L
for(pw in 0:(order-1L)){
pos_x = (uw_x+pw-1L)*n+seq1n
pos_1 = (uw_x+pw) == 1
pos_m = (uw_x+pw)==m
pos_other = pos_1==FALSE & pos_m==FALSE
# 1<u<m
M_psi_nm[pos_x[pos_other]]=(6*(x-Alpha_star[uw_x+pw])*(x-Alpha_star[uw_x+pw+order])/
((Alpha_star[uw_x+pw]-Alpha_star[uw_x+pw+order])^3))[pos_other]
# case u=1
M_psi_nm[pos_x[pos_1]]=(12*(x-Alpha_star[uw_x+pw+order])*(x-2*Alpha_star[uw_x+pw]+Alpha_star[uw_x+pw+order])/
((2*Alpha_star[uw_x+pw]-2*Alpha_star[uw_x+pw+order])^3))[pos_1]
# case u=m
M_psi_nm[pos_x[pos_m]]=(12*(x-Alpha_star[uw_x+pw])*(x+Alpha_star[uw_x+pw]-2*Alpha_star[uw_x+pw+order])/
((2*Alpha_star[uw_x+pw]-2*Alpha_star[uw_x+pw+order])^3))[pos_m]
}
M_psi_nm = M_psi_nm[,1:m,drop=FALSE]
3
if(which.matrix[2]){
rank.x = rank(x)
         = x[order(x)]
Alpha_x = sapply(x,function(y,lim=Alpha[-1L])sum(lim<y)+1L)</pre>
# integral equals 1
       = cumsum(tabulate(Alpha_x,n.Alpha-1))
up_u
for(uw in 1:(m-order+1)){M_Psi_nm[min(n,up_u[uw]+1):n,uw] = 1}
# other cases
if(basis=="msplines"){
Alpha_star2 = c(rep(Alpha[1],order),Alpha,rep(Alpha[n.Alpha],order))
factor_v = c((Alpha_star2[(order+2):length(Alpha_star2)]-Alpha_star2[1:(length(Alpha_star2)-order-1)])/
(order+1),rep(0,order-1))
M_psi2_nm = cbind(basis_mpl(x,knots,basis=basis,order=order+1,which=1),matrix(0,n,order-1))
pos_xo = rep((Alpha_x-1L)*n,1)+seq1n
pos_xo1 = rep(pos_xo,order)+rep(1:order,each=n)*n
for(ow in 0:(order-1)){
M_Psi_nm[pos_xo+ow*n] = apply(matrix(M_psi2_nm[pos_xo1+ow*n]*
factor_v[rep(Alpha_x,order)+rep((1:order)+ow,each=n)],ncol=order),1,sum)
}
}else{
Alpha_star_x = sapply(x,function(y,lim=Alpha[-1L])sum(lim<y)+1L)+order-1L</pre>
uw_x = Alpha_star_x-order+1L
for(pw in 0:(order-1L)){
pos_x = (uw_x+pw-1L)*n+seq1n
pos_1 = (uw_x+pw)==1
pos m = (uw x+pw) == m
pos_other = pos_1==FALSE & pos_m==FALSE
# 1<u<m
```

M_Psi_nm[pos_x[pos_other]]=((x-Alpha_star[uw_x+pw])^2*(2*x+Alpha_star[uw_x+pw]-3*Alpha_star[uw_x+pw+order])/ ((Alpha_star[uw_x+pw]-Alpha_star[uw_x+pw+order])^3))[pos_other] # case u=1 M_Psi_nm[pos_x[pos_1]]=((x-Alpha_star[uw_x+pw])*(x^2-2*x*Alpha_star[uw_x+pw]-2*Alpha_star[uw_x+pw]^2+ 6*Alpha_star[uw_x+pw]*Alpha_star[uw_x+pw+order]-3*Alpha_star[uw_x+pw+order]^2)/ (2*(Alpha_star[uw_x+pw]-Alpha_star[uw_x+pw+order])^3))[pos_1] # case u=m M_Psi_nm[pos_x[pos_m]]=((x-Alpha_star[uw_x+pw])^2*(x+2*Alpha_star[uw_x+pw]-3*Alpha_star[uw_x+pw+order])/ (2*(Alpha_star[uw_x+pw]-Alpha_star[uw_x+pw+order])^3))[pos_m] } } M_Psi_nm = M_Psi_nm[rank.x,1:m,drop=FALSE] }}} # pdf and cdf if(all(which.matrix)){list(psi=M_psi_nm,Psi=M_Psi_nm) # pdf or cdf }else{if(which.matrix[1]){M_psi_nm}else{M_Psi_nm}} } #~

A.2 Simulating Survival Data with TVC

In this section of the appendix, we discuss the simulation scheme of survival data with time-varying covariates. This simulation scheme is used for Paper 1 which focussed on survival analysis.

A.2.1 R Code

#	
# Step 1: Packages and Optoins	
# Libraries	
library(data.table)	
library(sqldf)	
library(survival)	
library(tidyverse)	
library(reshape2)	
library(Matrix)	
library(stringr)	
library(tcltk)	
# No scientific noitation	
options(scipen=999)	
¥	
"	
#	
# Step 2: FUNCTION quickSim() to draw 1 sample	
Sim <- function(sampSize, in.t.naught, n.knots, range.quant, censor.ub, basis, neb){	
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
# Simulation using Austin (2012)	
# Generating survival times to simulate Cox proportional hazards	
# models with time-varying covariates	
<pre># https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3546387/</pre>	
#	
<pre># sampSize = size of survival sample to draw</pre>	
# in.t.naught = the time at which sunjects possible switch from $z=0$ to $z=1$	
<pre># n.events_basis = number of events in each basis fucntion for MLE</pre>	
<pre># censor.ub = upper bound for censoring distribution U(0,censor.ub)</pre>	
#~~~~~~~	
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
# Step 1: Covariates and parameters	
# Sample size	
n=sampSize	
# Continuous and binary baseline covariate	
x1=rnorm (n,mean=0,sd=1)	
x2=rbinom(n,size=1, prob=0.5)	
# Baseline and time-varying effects	
beta1=1;beta2=1;gamma=1	
xbeta=x1*betal +x2*beta2	
# z swaps values in the following manner 50% of subjects	
<pre>switch =runif(n, min=0, max=1)>0</pre>	
# Randomly adust the swtich time	

c1=in.t.naught[1] c2=in.t.naught[2] c3=in.t.naught[3] ct1=runif(n, min=c1*.95, max=c1*1.05) ct2=runif(n, min=c2*.95, max=c2*1.05) ct3=runif(n, min=c3*.95, max=c3*1.05) t1=ifelse(switch==TRUE,ct1,1e10) t2=ifelse(switch==TRUE,ct2,1e10) t3=ifelse(switch==TRUE,ct3,1e10) # Weibull distribution lambda = 1 # Scale parameter nu = 1.5 # Shape parameter #~~~~~~~~~~~~ #~~~~~~~~~~~ # Step 2: Inverst transform # Uniform(0.1) for probability inverse transform u=runif(n, min=0, max=1) #~~~ #~ # Step 3: Undertake time.event simulation # Dichotomous time-varying covariate with three changes val1.upr=-log(u) val1.lwr=(lambda*exp(xbeta)) val1=(val1.upr/val1.lwr)^(1/nu) val2.upr=-log(u) - lambda*exp(xbeta)*t1^nu + lambda*exp(xbeta+gamma)*t1^nu val2.lwr=lambda*exp(xbeta+gamma) val2=(val2.upr/val2.lwr)^(1/nu) val3.upr=-log(u) - lambda*exp(xbeta)*t1^nu - lambda*exp(xbeta+gamma)*(t2^nu-t1^nu) + lambda*exp(xbeta)*t2^nu val3.lwr=lambda*exp(xbeta) val3=(val3.upr/val3.lwr)^(1/nu) val4.upr=-log(u) - lambda*exp(xbeta)*t1^nu - lambda*exp(xbeta+gamma)*(t2^nu-t1^nu) lambda*exp(xbeta)*(t3^nu-t2^nu) + lambda*exp(xbeta+gamma)*t3^nu val4.lwr=lambda*exp(xbeta+gamma) val4=(val4.upr/val4.lwr)^(1/nu) cut0=0 cut1=lambda*exp(xbeta)*(t1^nu) cut2=lambda*exp(xbeta)*(t1^nu + exp(gamma)*(t2^nu-t1^nu)) cut3=lambda*exp(xbeta)*(t1^nu + exp(gamma)*(t2^nu-t1^nu) + (t3^nu-t2^nu)) cut4=1e10 R1=ifelse(-log(u)>= cut0 & -log(u)< cut1, val1,0)</pre> R2=ifelse(-log(u)>= cut1 & -log(u)< cut2, val2,0)</pre> R3=ifelse(-log(u)>= cut2 & -log(u)< cut3, val3,0) R4=ifelse(-log(u)>= cut3 & -log(u)< cut4, val4,0) time.event=pmax(R1,R2,R3,R4) #~~ #~~ # Step 4: Undertake time.censor simulation time.censor=runif(n=n, min=0, max=censor.ub) #~~~ #~~~ # Step 5: Time and status time =ifelse(time.event<=time.censor, time.event, time.censor)</pre>

statusA=ifelse(time.event<=time.censor, 1, 0)</pre>

#~~~~~~

```
#~~~~~~~~
# Step 6: Modelling data
df=data.frame(id=seq(1,n,1), time=time, statusA=statusA, x1=x1, x2=x2)
df.longa =data.frame(id=df$id, time=0, zz=0)
df.longb =data.frame(id=df$id, time=ct1, zz=1)
df.longc =data.frame(id=df$id, time=ct2, zz=0)
df.longd =data.frame(id=df$id, time=ct3, zz=1)
df.longabcd=bind_rows(df.longa, df.longb, df.longc, df.longd) %>% arrange(id)
# Use tmerge
aaa<-tmerge(df,df,id=id,</pre>
status=event(time,statusA))
# Use tmerge again
bbb <- tmerge(aaa,df.longabcd,id=id, finalz=tdc(time,zz))</pre>
ccc <- tmerge(bbb,df.longabcd,id=id, finalz=tdc(time,zz))</pre>
df.long <- tmerge(ccc,df.longabcd,id=id, finalz=tdc(time,zz))</pre>
censorRate=1-mean(df$statusA)
#~~~
#~~~~~~~~
# Step 8: PL estimation
plFit=coxph(Surv(tstart, tstop, status)~ x1 + x2 + finalz + cluster(id),
data=df.long)
plCoef=coef(plFit)
#~~~~~~~~~~~~~~~~~~
#~~~~~~~~~~
# Step 9: ML estimation
mlFit=coxph_mlt(Surv(time, statusA) ~ x1 + x2,
data=df,
formula.z=Surv(tstart, tstop, status)~ finalz,
riji=tstop,
subject=id,
data.z=df.long,
max.iter=c(50,2000),
basis=basis,
n.events_basis=neb,
n.knots=n.knots,
range.quant=range.quant,
order=3)
mlCoef=mlFit$coef
#~~~~~~~~~~~
#~~~~
# Step 10: Collate output list
final=list(df=df,
df.long=df.long,
censorRate=censorRate,
plFit=plFit,
plCoef=plCoef,
plCoefSE=sqrt(diag(plFit$var)),
mlCoef=mlCoef,
mlCoefSE=mlFit$matricies$se.Eta_H,
mlFit=mlFit)
return(final)
#~~
}
#-----
```

A.3 Simulating Competing Risk Data with TVC

This section devises a scheme to simulate competing risk data for two competing risks, each with different Weibull baseline cause-specific hazards. This simulation scheme is used for Paper 2 which focussed on competing risk analysis.

A.3.1 Background

Figure A.1 displays the setting where all subjects begin in the same state and can encounter one fo two mutually exclusive events. More formally, let X_t be the state a subject is in at time t (t > 0), so that $X_t \in 0, 1, 2$. All subjects begin in state 0 at time t = 0 ($X_0 = 0$) and persist there until transitioning at time t = T to either event 1 ($X_T = 1$) or event 2 ($X_T = 2$).



Figure A.1: Competing Risk Analysis Example of 2 Possible Events

While there are several different parametrisations of the Weibull distribution, we adopt the parametrisation used in Bender et al. (2005) who define the baseline hazard

$$h_0(t) = \lambda v t^{\nu - 1} \tag{A.3}$$

where ν and λ are the shape and scale parameters respectively. Others, such as Klein and Moeschberger (2003) and Beyersmann et al. (2012) use the alternate parametrisation of $\nu = a$ and $\lambda = b^{-a}$; this is also in keeping with the parametrisation used in the R function rweibull.

The cause-specific hazards baseline covariates (x_1 and x_2) and a single time-varying covariate (z(t)) are

$$h_1(t) = h_{01}(t)exp(\beta_{11}x_1 + \beta_{12}x_2 + \gamma_1 z(t))$$
(A.4)

$$h_2(t) = h_{02}(t)exp(\beta_{21}x_1 + \beta_{22}x_2 + \gamma_2 z(t))$$
(A.5)

where β_{11} , β_{12} , γ_1 and β_{21} , β_{22} , γ_2 are the regression coefficients for the first and second competing risks respectively. For each cause-specific hazard, the Weibull baseline hazards are $h_{01}(t) = \lambda_1 v_1 t^{v_1-1}$ and $h_{02}(t) = \lambda_2 v_2 t^{v_2-1}$, where v_k and λ_k are the shape and scale parameters respectively for the k = 1, 2 competing risks. The all-risk hazard describes the hazard for any event

$$h(t) = h_{01}(t)exp(\beta_{11}x_1 + \beta_{12}x_2 + \gamma_1z(t)) + h_{02}(t)exp(\beta_{21}x_1 + \beta_{22}x_2 + \gamma_2z(t))$$
(A.6)

and the associated all-risk cumulative hazard is

$$H(t) = \int_0^t h_{01}(s) exp(\beta_{11}x_1 + \beta_{12}x_2 + \gamma_1 z(s)) + h_{02}(s) exp(\beta_{21}x_1 + \beta_{22}x_2 + \gamma_2 z(s)) ds.$$
(A.7)

For the Weibull baseline hazards, this results in

$$H(t) = \int_{0}^{t} \lambda_{1} v_{1} s^{v_{1}-1} exp(\beta_{11}x_{1} + \beta_{12}x_{2} + \gamma_{1}z(s))ds + \int_{0}^{t} \lambda_{2} v_{2} s^{v_{2}-1} exp(\beta_{21}x_{1} + \beta_{22}x_{2} + \gamma_{2}z(s))ds H(t) = \lambda_{1} v_{1} exp(\beta_{11}x_{1} + \beta_{12}x_{2}) \int_{0}^{t} s^{v_{1}-1} exp(\gamma_{1}z(s))ds + \lambda_{2} v_{2} exp(\beta_{21}x_{1} + \beta_{22}x_{2}) \int_{0}^{t} s^{v_{2}-1} exp(\gamma_{2}z(s))ds.$$
(A.8)

A.3.2 Simulating Survival Time

We define z(t) as a dichotomous time-varying covariate with at most one change from untreated (z = 0) to treated (z = 1) and that this switch (should it occur for a subject) does so at the time $t = t_1$. This means we need to evaluate the cumulative hazard in the integral in equation (A.8) for the following two time domains

- z = 0, for $t < t_1$
- z = 1, for $t \ge t_1$.

Separating the integrals in equation (A.8) into these two time domains results in

$$H(t) = \lambda_{1}v_{1}exp(\beta_{11}x_{1} + \beta_{12}x_{2}) \left[\int_{0}^{t_{1}} s^{\nu_{1}-1}ds + \int_{t_{1}}^{t} s^{\nu_{1}-1}exp(\gamma_{1})ds \right] + \lambda_{2}v_{2}exp(\beta_{21}x_{1} + \beta_{22}x_{2}) \left[\int_{0}^{t_{1}} s^{\nu_{2}-1}ds + \int_{t_{1}}^{t} s^{\nu_{2}-1}exp(\gamma_{1})ds \right] H(t) = \lambda_{1}v_{1}exp(\beta_{11}x_{1} + \beta_{12}x_{2}) \left[\frac{1}{\nu_{1}}t_{1}^{\nu_{1}} + \frac{1}{\nu_{1}}t^{\nu_{1}}exp(\gamma_{1}) - \frac{1}{\nu_{1}}t_{1}^{\nu_{1}}exp(\gamma_{1}) \right] + \lambda_{2}v_{2}exp(\beta_{21}x_{1} + \beta_{22}x_{2}) \left[\frac{1}{\nu_{2}}t_{1}^{\nu_{2}} + \frac{1}{\nu_{2}}t^{\nu_{2}}exp(\gamma_{2}) - \frac{1}{\nu_{2}}t_{1}^{\nu_{2}}exp(\gamma_{2}) \right].$$
(A.9)

This implies

$$H(t) = \begin{cases} \lambda_1 exp(\beta_{11}x_1 + \beta_{12}x_2)t^{\nu_1} + \lambda_2 exp(\beta_{21}x_1 + \beta_{22}x_2)t^{\nu_2} &, \text{ if } t < t_1 \\ \lambda_1 exp(\beta_{11}x_1 + \beta_{12}x_2) \begin{bmatrix} t_1^{\nu_1} + t^{\nu_1} exp(\gamma_1) - t_1^{\nu_1} exp(\gamma_1) \\ t_1^{\nu_2} + t^{\nu_2} exp(\gamma_2) - t_1^{\nu_2} exp(\gamma_2) \end{bmatrix} &, \text{ if } t \ge t_1. \end{cases}$$

(A.10)

This partitions the domain of the cumulative hazard into two intervals: $D_1 = (0, t_1)$ and $D_2 = [t_1, \infty)$. Let

•
$$r_1 = \lambda_1 exp(\beta_{11}x_1 + \beta_{12}x_2)t_1^{\nu_1} + \lambda_2 exp(\beta_{21}x_1 + \beta_{22}x_2)t_1^{\nu_2}$$

so that

•
$$R_1 = (0, r_1)$$

• $R_1 = [r_1, \infty)$

are the ranges of the cumulative hazard function associated with domains D_1 and D_2 . In order to simulate using this scheme, we need to invert H(t) in equation (A.10). There are two approaches:

- 1. adopt the simplification $\lambda_1 = \lambda_2 = \lambda$ and $\nu_1 = \nu_2 = \nu$ and invert analytically; or
- 2. invert numerically.

A.3.2.1 Analytical Inversion of H(t)

For the first approach, we analytically invert H(t) from equation (A.10). For the domain D_1 , the inverse is

$$H^{-1}(t) = \left(\frac{t}{\lambda \left(exp(\beta_{11}x_1 + \beta_{12}x_2) + exp(\beta_{21}x_1 + \beta_{22}x_2)\right)}\right)^{1/\nu}$$
(A.11)

and analytic inverse inverse of H(t) for the domain D_2 is

$$H^{-1}(t) = \left(\frac{t - \lambda t_1^{\gamma} \left(exp(\beta_{11}x_1 + \beta_{12}x_2) + exp(\beta_{21}x_1 + \beta_{22}x_2)\right) + \lambda t_1^{\gamma} \left(exp(\beta_{11}x_1 + \beta_{12}x_2)exp(\gamma_1) + exp(\beta_{21}x_1 + \beta_{22}x_2)exp(\gamma_1)\right)}{\lambda \left(exp(\beta_{11}x_1 + \beta_{12}x_2)exp(\gamma_1) + exp(\beta_{21}x_1 + \beta_{22}x_2)exp(\gamma_2)\right)}\right)^{1}$$
(A.12)

Therfor we can simulate survival time as

$$T = \begin{cases} \left(\frac{-ln(u)}{\lambda\left(exp(\beta_{11}x_{1}+\beta_{12}x_{2})+exp(\beta_{21}x_{1}+\beta_{22}x_{2})\right)}\right)^{1/\nu} , \text{ if } -log(u) < r_{1} \\ \left(\frac{-ln(u)-\lambda t_{1}^{\nu}\left(exp(\beta_{11}x_{1}+\beta_{12}x_{2})+exp(\beta_{21}x_{1}+\beta_{22}x_{2})\right)+\lambda t_{1}^{\nu}\left(exp(\beta_{11}x_{1}+\beta_{12}x_{2})exp(\gamma_{1})+exp(\beta_{21}x_{1}+\beta_{22}x_{2})exp(\gamma_{1})\right)}{\lambda\left(exp(\beta_{11}x_{1}+\beta_{12}x_{2})exp(\gamma_{1})+exp(\beta_{21}x_{1}+\beta_{22}x_{2})exp(\gamma_{2})\right)}\right)^{1/\nu} , \text{ if } -log(u) \geq r_{1} \end{cases}$$
(A.13)

where $u \sim Uni(0, 1)$.

A.3.2.2 Numerical Inversion of H(t)

For the second approach, we numerically invert H(t) from equation (A.10). For a given y and H(t), we wish to compute t such that H(t) = y. To achieve this, we use the stats::uniroot() function to fine the roots of the equation H(t) - y = 0, evaluated at the value y = -ln(u).

A.3.3 Simulating Event Type

To simulate which event occurs, we return to figure A.1. The probability that the event which occurs at time t = T is event 1 ($X_T = 1$) is

$$P[X_T = 1|t < T \le t + dt, T \ge t] = \frac{P[t < T \le t + dt, X_T = 1]}{P[T \le t + dt|T \ge t]}$$
$$= \frac{h_{01}(t)}{h_{01}(t) + h_{02}(t)}$$
$$= \frac{\lambda_1 \nu_1 t^{\nu_1 - 1}}{\lambda_1 \nu_1 t^{\nu_1 - 1} + \lambda_2 \nu_2 t^{\nu_2 - 1}}.$$
(A.14)

To simulate, we draw from the binomial distribution $x_T \sim Bin(p)$, where the parameter p is equation (A.14).

A.3.4 R Code

```
# Step 1: Packages and Options
# # Packages
library(Matrix)
library(ggplot2)
library(plyr)
library(tidyverse)
library(survival) # 2.44-1.1
library(data.table);library(broom);library(cmprsk)
# Options
#No scientific noitation
options(scipen=999)
                    #====
          _____
# Step 2: FUNCTION inverse.eval() - Evaluate the inverse of a function at a specified value
inverse.eval = function (inputFun, x.inverse.eval, lower = -100, upper = 10000) {
# Numerical inverse of a function
# https://stackoverflow.com/questions/10081479/solving-for-the-inverse-of-a-function-in-r
# Solve the roots (ie: the x-intercepts) for: f(x) - y = 0
# inputFun - the function to numerically invert
# x.inverse.eval - a vector of x-values to evaluate the numerical inverted function
# lower, upper - ranges to pass to uniroot()
#~~
# Step 2A: Initialise empty output
y.inv.numeric.return=as.matrix(rep(0,length(x.inverse.eval)))
# Step 2B: Loop over each input x-value
for (i in 1:length(x.inverse.eval)){
# Step 2Bi: Extract the ith x-value
this.x.inverse.eval=x.inverse.eval[i]
```

A.3 Simulating Competing Risk Data with TVC

```
# Step 2Bii: Call uniroot() to solve f(x) - y = 0
uniroot.arg1 = uniroot((function (internal.x=lower) {inputFun(internal.x) - this.x.inverse.eval}),
lower = lower, upper = upper)[1]
# Step 2Biii: Save the ith value the inverted function. as evaluated at x[i]
y.inv.numeric.return[i]=as.numeric(uniroot.arg1)
}
# Step 2C: Return the results
return(y.inv.numeric.return)
}
#===
#-----
# Step 3: FUNCTION simCall() - Simulate competing risk survuval time
simCall <- function(n,</pre>
lambda1, lambda2,
nu1, nu2,
x1, x2,
t.one,
beta11, beta12, gamma1,
beta21, beta22, gamma2){
#~~~~
# Simulate competing risk survival time
# lambda1, lambda2
                       = Scale parameters
# nu1,nu2 = Shape parameters
# t.one = the value of t at which z switches from 0 to 1
# beta11, beta12, gamma1 = regression parameters from first CR
# beta21, beta22, gamma2 = regression parameters from second CR
#~~~~
#~~
# Step 3A: Inverst transform
# Uniform(0,1) for probability inverse transform
u=runif(n, min=0, max=1)
#~~~~
#~~
# Step 3B: Linear predictors
xbeta1=x1*beta11 +x2*beta12
xbeta2=x1*beta21 +x2*beta22
#~~
# Step 3C: H(t) functions that need to be numerically inverted
Hr1 = function(t){
value = lambda1 * exp(xbeta1.value) * t^nu1 +
lambda2 * exp(xbeta2.value) * t^nu2
return(value)
}
Hr2 = function(t){
value = lambda1 * exp(xbeta1.value) * (t.one.value^nu1 + t^nu1 * exp(gamma1) - t.one.value^nu1 * exp(gamma1)) +
lambda2 * exp(xbeta2.value) * (t.one.value^nu2 + t^nu2 * exp(gamma2) - t.one.value^nu2 * exp(gamma2))
return(value)
3
#~~~
# Step 3D: Loop over each value for the inversion
event.time.num=rep(0,n)
for (i in 1:n){
# Extract the ith value from each vector
```

```
xbeta1.value=xbeta1[i]
xbeta2.value=xbeta2[i]
t.one.value=t.one[i]
neg.log.u=-log(u)[i]
# Test which piece-wise function to invert
#test=(-log(u)<(lambda1 * exp(xbeta1.value) * t.one^nu1 +</pre>
             lambda2 * exp(xbeta2.value) * t.one^nu2))
#
test.num=(neg.log.u<Hr1(t.one.value))</pre>
# Undertake the correct inversion
if (test.num==FALSE){
temp=inverse.eval(inputFun=Hr2, x.inverse.eval=neg.log.u, lower=1E-12)
} else
ł
temp=inverse.eval(inputFun=Hr1, x.inverse.eval=neg.log.u, lower=1E-12)
}
# Save the results
event.time.num[i]=temp
}
#~~~
#~~
# Step 3D: Simulate event.type.num
h01.num=lambda1 * nu1 * event.time.num^{nu1-1}
h02.num=lambda2 * nu2 * event.time.num^{nu2-1}
p.num=h01.num / (h01.num + h02.num)
event.type.num=rbinom(n=n, p=p.num, size=1)+1
#~~~~~
#~~~~~
# Step 3E: Combine results
all=bind_cols(event.time=event.time.num, event.type=event.type.num)
return(all)
#~~~
}
#~~
#-----
#==
# Step 4: FUNCTION simAndFit()
simAndFit <- function(sampSize, in.t.naught, censor.ub, pct, neb, man, 1, u){</pre>
#~~
# Simulate survival times and fit Cox proportional hazards models with time-varying covariates
# https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3546387/
# sampSize
            = sample size to draw
# in.t.naught = the times at which subjects possible switch from z=0 to z=1
# n.knots
               = c(v1, v2) nknots call to coxph_mlt
# range.quant = range.quant() call to coxph_mlt
# censor.ub = upper bound for censoring distribution U(0, censor.ub)
# basis
                = basis for baseline hazard
# pct
               = percent of observations that switch
# neb
               = number of events in each basis function
# man
               = 1 for manual knots, 0 otherwise
# 1
                = lambda
                = nu
# u
#~~
#~~~
# Step 4A: Covariates and parameters
```

n=sampSize

#~~/

Continuous and binary baseline covariate
x1=rnorm (n,mean=0,sd=1)
x2=rbinom(n,size=1, prob=0.5)
Baseline and time-varying effects
beta11=1;beta12=1;gamma1=1
beta21=1;beta22=1;gamma2=1

Step 4B: z swaps from z=0 to z=1 at time=5 for 100% of subjects #switch =runif(n, min=0, max=1)>0 switch =runif(n, min=0, max=1)>(1-pct)

Randomly adust the swtich time c2=in.t.naught ct2=runif(n, min=c2*.95, max=c2*1.05) t2=ifelse(switch==TRUE,ct2,le10)

Weibull distribution
lambda1=1
lambda2=1
nu1=u
nu2=u
#

#~~~~~~

Step 4C: Draw the simulation # Only use t2 as the single cut point for now parms=data.frame(lambda1=lambda1, lambda2=lambda2, nu1=nu1, nu2=nu2) sim=simCall(n=n, lambda1=lambda1, lambda2=lambda2, nu1=nu1, nu2=nu2, x1=x1, x2=x2, t.one=t2, beta11=beta11, beta12=beta12, gamma1=gamma1, beta21=beta21, beta22=beta22, gamma2=gamma2)

#~~

#~~~~ # Step 4E: Time and status time.df0 = data.frame(sim, time.censor, switch.time=t2, x1=x1, x2=x2) %>% mutate(id=row_number()) %>% mutate(time =ifelse(event.time<=time.censor, event.time, time.censor)) %>% mutate(status=ifelse(event.time<=time.censor, event.type, 0)) %>% # TVC mutate(actual.switch =ifelse(time>switch.time,T,F), actual.switch.time=ifelse(time>switch.time,switch.time,NA), =ifelse(time>switch.time,1,0)) %>% z # Make individual statuses mutate(status1=ifelse(status==1,1,0), status2=ifelse(status==2,1,0)) %>% select(id, time, status, status1, status2, x1, x2, actual.switch, actual.switch.time, z) time.df = time.df0 #~~

df =time.df %>% select(id, time, actual.switch.time, status, x1, x2, z) # Ready for tmerge() df.longa =data.frame(id=df\$id, time=0, zz=0) df.longb =data.frame(id=df\$id, time=df\$actual.switch.time, zz=1) df.longab=bind_rows(df.longa, df.longb) %>% arrange(id) # Use tmerge aaa<-tmerge(df,df,id=id,</pre> status=event(time,status)) # Use tmerge again temp_df.long <- tmerge(aaa,df.longab,id=id,</pre> finalz=tdc(time,zz)) # Make individual statuses df.long = temp_df.long %>% mutate(status1=ifelse(status==1,1,0), status2=ifelse(status==2,1,0)) #~~~ #~-# Split the baseline datasets df_1 = time.df %>% select(id, x1, x2, time, status1) %>% as.data.frame df_2 = time.df %>% select(id, x1, x2, time, status2) %>% as.data.frame # Split the baseline datasets df_long_1 = df.long %>% select(id, x1, x2, finalz, tstart, tstop, status1) %>% as.data.frame df_long_2 = df.long %>% select(id, x1, x2, finalz, tstart, tstop, status2) %>% as.data.frame #~~~ #~~~~ # Censor Rate censorRate=mean(df\$status==0) # z rate zRate=sum(df\$z)/n # Status rate statusRate=as.data.frame(table(df\$status)) #~~ #~~ # Step 4G: PL estimation for event 1 plFit_1=coxph(Surv(tstart, tstop, status1)~ x1 + x2 + finalz, data=df_long_1, ties='breslow') plCoef_1=coef(plFit_1) #~~~ #~~~ # Step 4I(2): PL estimation for event 2 plFit_2=coxph(Surv(tstart, tstop, status2)~ x1 + x2 + finalz, data=df_long_2, ties='breslow') plCoef_2=coef(plFit_2) #~~~~~~~~~~~~~ #~~ # Step 4I(3): collate the PL results temp1a=bind_cols(Risk=rep(1,3), Type=rep('Coef',3), Parm=c('x1', 'x2', 'z1'), Coef=plCoef_1)

temp1b=bind_cols(Risk=rep(1,3), Type=rep('SE',3), Parm=c('x1', 'x2', 'z1'), Coef=sqrt(diag(plFit_1\$var)))

temp2a=bind_cols(Risk=rep(2,3), Type=rep('Coef',3), Parm=c('x1', 'x2', 'z1'), Coef=plCoef_2)

A.3 Simulating Competing Risk Data with TVC

temp3 =bind_cols(Risk=rep(0,2), lype=rep('0th',2), Pari	<pre>m=c('censorRate','zRate'), Coef=c(censorRate, zRate))</pre>
both_plCoef=bind_rows(temp1a, temp1b, temp2a, temp2b, temp	emp3)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	
#~~~~~~~~~~~	
if (man==1){	
<pre>kman1=c(quantile(df_1\$time, probs=seq(0,1,0.1)))</pre>	
<pre>kman2=c(quantile(df_2\$time, probs=seq(0,1,0.1)))</pre>	
} else{	
kman1=NULL	
kman2=NULL	
}	
#~~~~~~~~~~~~	
#NNNNNNNNNNNNNNNNNN	
<pre># Step 4J(1): nL estimation for event 1 mlFit 1=count mlt(Sum(time_status1) = 1 + m2</pre>	
$m_{\text{III}} = -\cos pn_{\text{III}} = \cos pn_{\text{IIII}} = \cos pn_{\text{IIIII}} = \cos pn_{\text{IIIII}} = \cos pn_{\text{IIIIII}} = \cos pn_{\text{IIIIIIIIIIII}} = \cos pn_{IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII$	
formula z- Surv(tetart teton etatuel) finalz	
riji-teton	
subject=id	
data $z=df \log 1$	
max iter= $c(50, 2000)$	
basis='uniform'.	
knots.manual=kman1.	
n.events basis=neb)	
#~~~~~~~	
#~~~~~~~~~	
<pre># Step 4J(2): ML estimation for event 2</pre>	
<pre>mlFit_2=coxph_mlt(Surv(time, status2) ~ x1 + x2,</pre>	
data=df_2,	
formula.z= Surv(tstart, tstop, status2) ~ finalz,	
riji=tstop,	
subject=id,	
data.z=df_long_2,	
max.iter=c(50,2000),	
basis='uniform',	
knots.manual=kman2,	
n.events_basis=neb)	
#~~~~~~~	
#~~~~~~~~~~	
# Step 4K: Collate output list	
<pre>final=list(censorRate=censorRate,</pre>	
zRate=zRate,	
statusRate=statusRate,	
parms=parms,	
sim=sim,	
df=df,	
df.long=df.long,	
df_1=df_1,	
df_2=df_2,	
df_long_1=df_long_1,	
df_long_2=df_long_2,	
plFit_1=plFit_1,	
plFit_2=plFit_2,	
<pre>plCoefSE_1=sqrt(diag(plFit_1\$var)),</pre>	
<pre>plCoefSE_2=sqrt(diag(plFit_2\$var)),</pre>	
mlFit_1=mlFit_1,	
mlFit_2=mlFit_2,	
<pre>mlCoefSE_1=t(t(mlFit_1\$matricies\$se.Eta_H)),</pre>	

return(final)

#~~~~~~

} #==

A.4 Model Implementation - Mixture-Cure Model

In this section of the appendix, we discuss the model implementation of the mixture-cure model. This implementation is used for both Paper 3 which focussed mixture-cure analysis.

A.4.1 R Code



```
# W matrix
var.w=all.names(formula.cure)
temp1 = var.w != '~'
temp2 = var.w != '+'
temp3 = var.w != 'formula.cure'
temp4 = temp1 * temp2 * temp3 * seq(1,length(var.w))
var.w = var.w[temp4]
Wtemp=data[,var.w]
temp=rep(1,n)
Intercept=data.frame(Intercept=temp)
W=data.frame(Intercept,Wtemp)
W=as.matrix(W)
q=ncol(W)
# Knot sequence and psi matrices
knots = knots_smcml(control, t_i)
     = knots$m
m
# Basis functions
psi = basis_smcml(t_i,knots,control$basis,control$order,which=1)
PSI = basis_smcml(t_i,knots,control$basis,control$order,which=2)
#===
#=========
# Fit the model
Beta = as.matrix(rep(0,p))
Alpha
       = as.matrix(rep(0,q))
Theta
         = as.matrix(rep(1,knots$m))
fit <- smcmlfit(status=status,</pre>
X=X,
W=W
psi=psi,
PSI=PSI,
Beta0=Beta,
Theta0=Theta,
Alpha0=Alpha,
kappa=control$kappa,
convVal=control$tol,
minTheta=control$epsilon,
maxiter=control$max.iter)
#=================
#==
# Identify theta values that are zero
pos = c(rep(TRUE,p), rep(TRUE,q), !(abs(fit$coef$Theta)<1E-5 & (fit$GradTheta< -1E-2)))</pre>
#===
#=====
# Calculate Hessian
Hessian = as.matrix(fit$matricies$Hessian)
HessianInv = matrix(0,p+q+m,p+q+m)
temp = try(solve(-Hessian[pos,pos]),silent=TRUE)
if(class(temp)!="try-error"){
HessianInv[pos,pos] = temp
cov_NuNu_H = HessianInv
se.Eta_H
                = suppressWarnings(sqrt(diag(cov_NuNu_H)))
}else{
cov_NuNu_H = matrix(NA,p+q+m,p+q+m)
se.Eta_H = rep(NA,p+q+m)
}
```
```
# output
fit$matricies$cov_NuNu_H=cov_NuNu_H
fit$matricies$se.Eta_H=se.Eta_H
fit$knots = knots
fit$control = control
fit$call = match.call()
fit$dim = list(n = n, p = p, q = q, m = knots$m)
fit$data = list(time = t_i, observed = observed, X = X, W = W, name = data.name)
class(fit) = "smcml"
fit
#_____
}
#~
#~~
# Smcmlfit
smcmlfit <- function(status, X=X, W=W, psi=psi, PSI=PSI, Beta0, Theta0, Alpha0,</pre>
kappa, convVal, minTheta, maxiter){
# Initialise
p = length(Beta0)
q = length(Alpha0)
m = length(Theta0)
1 = p+m+q
n = nrow(X)
Beta=Beta0
Alpha=Alpha0
Theta=Theta0
#======
#===
# Function to update calculatoin of the log-likelihood
ll=function(thisBeta, thisAlpha, thisTheta, X, W, psi, PSI){
h0 = psi %*% thisTheta  # nxm x mx1 = nx1
H0 = PSI %*% thisTheta
                          # nxm x mx1 = nx1
H=H0*exp(X %*% thisBeta)  # nxp x px1 = nx1; element-wise nx1
S=exp(-H)
                          # element-wise nx1
pii=1/(1+exp(-W%*%thisAlpha)) # nxq x qx1; element-wise nx1
logliki = status * (log(pii) + log(h0) + X %*% thisBeta - H) + (1 - status) *log(pii * S + (1-pii))
loglik=sum(logliki)
return=list(h0=h0, H0=H0, h=h, H=H, S=S, pii=pii, loglik=loglik)
return(return)
fn=ll(Beta0, Alpha0, Theta0, X=X, W=W, psi=psi, PSI=PSI)
loglik0=fn$loglik
#===
# Save values for each iteration
this.maxiter=maxiter
loglikMat=matrix(rep(0,(this.maxiter+1)*3), ncol=3, nrow=this.maxiter+1)
loglikMat[1,]=loglik0
BetaMat =matrix(rep(0,(this.maxiter+1)*p), nrow=this.maxiter+1, ncol=p)
AlphaMat=matrix(rep(0,(this.maxiter+1)*q), nrow=this.maxiter+1, ncol=q)
ThetaMat=matrix(rep(0,(this.maxiter+1)*m), nrow=this.maxiter+1, ncol=m)
BetaMat [1,]=Beta0
AlphaMat[1,]=Alpha0
ThetaMat[1,]=Theta0
```

```
#____
for(iter in 1:maxiter){
# Update beta
# Grad Beta
Blvec = - status*fn$H - (1-status)*fn$H*(fn$pii*fn$S)/(fn$pii*fn$S+1-fn$pii)
GradBeta <- t(X) %*% (status + B1vec)</pre>
# Hessian Beta
B2vec = B1vec - (1-status)*fn$H*fn$H*fn$S*fn$pii*(1-fn$pii) / (fn$pii*fn$S + 1-fn$pii)^2
HessianBeta <- as.matrix(crossprod(X, Diagonal(n=n, x=B2vec)) %*% X)</pre>
# Step Beta
StepBeta <- solve(HessianBeta) %*% GradBeta
Beta <- Beta0 - StepBeta
fn=ll(Beta, Alpha0, Theta0, X=X, W=W, psi=psi, PSI=PSI)
loglik = fn$loglik
# Adapt Newton step if needed
r=0
while(loglik < loglik0){</pre>
r=r+1
StepBeta = StepBeta/kappa
Beta <- Beta0 - StepBeta
fn=ll(Beta, Alpha0, Theta0, X=X, W=W, psi=psi, PSI=PSI)
loglik = fn$loglik
if (r>500) break
}
loalik0=loalik
loglikMat[1+iter,1]=loglik
# Update Alpha
# Grad Alpha
Alvec = status*(1-fn$pii) + (1-status)*fn$pii*(1-fn$pii)*(fn$S-1)/(fn$pii*fn$S+1-fn$pii)
GradAlpha <- t(W) %*% (Alvec)</pre>
# Hessian Alpha
A2vec = - status*fn$pii*(1-fn$pii) - (1-status)* ( (fn$pii*(1-fn$pii)*(fn$S-1) ) / (fn$pii*fn$S+1-fn$pii) )^2
# Step Alpha
HessianAlpha <- as.matrix(crossprod(W, Diagonal(n=n, x=A2vec)) %*% W)</pre>
StepAlpha <- solve(HessianAlpha) %*% GradAlpha</pre>
#if(iter==30){browser()}
Alpha <- Alpha0 - StepAlpha
fn=ll(Beta, Alpha, Theta0, X=X, W=W, psi=psi, PSI=PSI)
loglik = fn$loglik
# Adapt Newton step if needed
r=0
while(loglik < loglik0){</pre>
r=r+1
StepAlpha = StepAlpha/kappa
Alpha <- Alpha0 - StepAlpha
```

fn=ll(Beta, Alpha, Theta0, X=X, W=W, psi=psi, PSI=PSI)

```
loglik = fn$loglik
if (r>500) break
}
loglik0=loglik
loglikMat[1+iter,2]=loglik
#===========
# Update theta
C=psi # nxm
Cstar=PSI # nxm
# Grad Theta
D=Diagonal(n=n, x=fn$h0) # nxn
f1=status*exp(X%*%Beta) + (1-status)*exp(X%*%Beta) * fn$pii * fn$S /(fn$pii*fn$S+1-fn$pii) #nx1
w u=t(Cstar) %*% f1 # mxn x nx1 = mx1
v_u=Theta0 / (w_u + 0) # element-wise mx1 / mx1 = mx1
StepTheta=v u * GradTheta # element-wise mx1 / mx1 = mx1
Theta = as.matrix(Theta0 + StepTheta)
Theta[which(Theta<minTheta)]=minTheta
fn=ll(Beta, Alpha, Theta, X=X, W=W, psi=psi, PSI=PSI)
loglik = fn$loglik
# Adapt Newton step if needed
r=0
while(loglik < loglik0){</pre>
r=r+1
StepTheta = StepTheta/kappa
Theta = as.matrix(Theta0 + StepTheta)
Theta[which(Theta<minTheta)]=minTheta
fn=ll(Beta, Alpha, Theta, X=X, W=W, psi=psi, PSI=PSI)
loglik = fn$loglik
if (r>500) break
}
loglik0=loglik
loglikMat[1+iter,3]=loglik
# Check for convergence
varepsilon <- max(c(abs(Alpha-Alpha0),abs(Beta-Beta0),abs(Theta-Theta0)))</pre>
if (varepsilon<convVal) break
Beta0 <- Beta
Alpha0 <- Alpha
Theta0 <- Theta
BetaMat [1+iter,]=Beta0
AlphaMat[1+iter,]=Alpha0
ThetaMat[1+iter,]=Theta0
if ( (round(iter/10,6)-floor(iter/10)) == 0){
cat(iter, 'iterations...\n')
}else{
#cat(iter, '\n')
}
cat(iter, 'iterations...\n')
#=====
#============
```

#Inference for Beta pxp Blvec = - status*fn\$H - (1-status)*fn\$H*(fn\$pii*fn\$S)/(fn\$pii*fn\$S+1-fn\$pii) B2vec = B1vec - (1-status)*fn\$H*fn\$H*fn\$S*fn\$pii*(1-fn\$pii) / (fn\$pii*fn\$S + 1-fn\$pii)*2 V1 = as.matrix(crossprod(X, Diagonal(n=n, x=B2vec)) %*% X) colnames(V1)=rownames(V1)=NULL #================== # Inference for Alpha qxq A2vec = - status*fn\$pii*(1-fn\$pii) + (1-status)*fn\$pii*(1-fn\$pii)*(fn\$S-1) * ((fn\$pii*fn\$S+1-fn\$pii)*(1-2*fn\$pii) - (fn\$pii-fn\$pii^2)*(fn\$S-1)) / (fn\$pii*fn\$S+1-fn\$pii)^2 V2 = as.matrix(crossprod(W, Diagonal(n=n, x=A2vec)) %*% W) colnames(V2)=rownames(V2)=NULL #== # Inference for Theta mxm aa = -t(C) %*% Diagonal(n=n, x=as.numeric(status/fn\$h0^2)) %*% C bb = (1-status) * (exp(X%*%Beta))^2 * fn\$pii * (1-fn\$pii) * fn\$S/(fn\$pii*fn\$S+1-fn\$pii)^2 cc = t(Cstar) %*% Diagonal(n=n, x=as.numeric(bb)) %*% Cstar V3 = (aa+cc) #===== #=========== # Inference for d2l/dalpha dbeta qxp aa= -(1-status) * fn\$pii * (1 - fn\$pii) * fn\$S * fn\$H /(fn\$pii*fn\$S+1-fn\$pii)^2 V21 = t(W) %*% Diagonal(n=n, x=aa) %*% X V21 = as.matrix(V21) colnames(V21)=rownames(V21)=NULL V12=t(V21) #================== # Inference for d21/dalpha dtheta qxm aa= -(1-status) * fn\$pii * (1 - fn\$pii) * fn\$S * exp(X%*%Beta) /(fn\$pii*fn\$S+1-fn\$pii)^2 V23 = t(W) %*% Diagonal(n=n, x=aa) %*% Cstar V23 = as.matrix(V23)colnames(V23)=rownames(V23)=NULL V32=t(V23) #_____ #==== # Inference for d21/dbeta dtheta pxm aa1= -status*exp(X%*%Beta) aa2= -(1-status)*exp(X%*%Beta)*fn\$pii*fn\$S/(fn\$pii*fn\$S+1-fn\$pii) aa3= (1-status)*exp(X%%Beta)*fn\$H*fn\$S*fn\$pii*(1-fn\$pii)/(fn\$pii*fn\$S+1-fn\$pii)^2 aa4=aa1+aa2+aa3 V13 = t(X) %*% Diagonal(n=n, x=aa4) %*% Cstar V13 = as.matrix(V13) colnames(V13)=rownames(V13)=NULL V31=t(V13) #================= # H matrix row1=cbind(V1, V12, V13) row2=cbind(V21, V2, V23) row3=cbind(V31, V32, V3) Hessian=rbind(row1, row2, row3) condnoAlpha=kappa(HessianAlpha) #======= #========= #Output return(list(coef=list(Beta=Beta, Alpha=Alpha, Theta=Theta),

A.4 Model Implementation - Mixture-Cure Model

```
condnoAlpha=condnoAlpha.
GradTheta=as.matrix(GradTheta),
loglik=list(iter=iter, loglik=fn$loglik),
matricies=list(h0=fn$h0, H0=fn$H0, h=fn$h, H=fn$H, S=fn$S, pii=fn$pii, Hessian=Hessian),
History=list(loglikMat=loglikMat, BetaMat=BetaMat, AlphaMat=AlphaMat, ThetaMat=ThetaMat)))
#======
3
#~,
# Control
smcml.control <- function(n.obs=NULL, basis = "uniform", max.iter=2000, tol=1e-7,</pre>
n.knots = NULL, n.events_basis = NULL, range.quant = c(0.075,.9),
min.theta = 1e-10, order = 3L,
kappa = 1/.6, epsilon = 1e-50, knots.manual=NULL){
tol
            = ifelse(tol>0 & tol<1,tol,1e-7)
           = ifelse(order>0 & order<6,as.integer(order),3L)
order
min.theta = ifelse(min.theta>0 & min.theta<1e-3,min.theta,1e-10)</pre>
          = ifelse(kappa>1, kappa, 1/.6)
kappa
if(all(range.quant<1) & all(range.quant>0) & length(range.quant)==2){
range.quant = range.quant[order(range.quant)]
else{range.quant = c(0.075,.9)
if(is.null(n.knots)|sum(n.knots)<3|length(n.knots)!=2){
n.knots
         = if(basis!='uniform' & basis!='msplines'){c(0,20)}else{c(8,2)}
if(!is.null(n.events_basis)){
n.events_basis = ifelse(n.events_basis<1|n.events_basis>floor(n.obs/2),
max(round(3.5*log(n.obs)-7.5),1L),round(n.events_basis))
}else{n.events_basis = max(round(3.5*log(n.obs)-7.5),1L)}
out = list(basis = basis, max.iter = max.iter, tol = tol,
n.knots = n.knots, range.quant = range.quant,
n.events_basis = as.integer(n.events_basis), min.theta = min.theta,
kappa = kappa, epsilon = epsilon,
knots.manual=knots.manual)
class(out) = "smcml.control"
out
}
#~
#~
# Knots
knots_smcml=function(control,events){
n.events = length(events)
range = range(events)
if(!(is.null(control$knots.manual))){
Zeta=c(control$knots.manual, range)
Zeta = Zeta[order(Zeta)]
Zeta=unique(sort(Zeta))
}else{
Zeta1 = quantile(events,seq(0,control$range.quant[2],length.out=(control$n.knots[1]+1)))
Zeta2 = seq(quantile(events,control$range.quant[2]),range(events)[2],length=control$n.knots[2]+2)
Zeta
      = c(Zeta1,Zeta2[-1])
}
n.Zeta = length(Zeta)
m
         = length(Zeta)-1
Delta
         = Zeta[2L:(m+1L)]-Zeta[1L:m]
list(m=m,Zeta=Zeta,Delta=Delta)
}
#~
# Basis
basis_smcml = function(x,knots,basis,order,which=c(1,2)){
```

```
which.matrix = rep(T,2)
which.matrix[-which]=FALSE
n = length(x)
Zeta = knots$Zeta
Delta = knots$Delta
n.Zeta = length(Zeta)
m = knots$m
M_Psi_nm = M_psi_nm = matrix(0,n,m)
u_i = sapply(x,function(y,lim=Zeta[-1L])sum(lim<y)+1L)</pre>
for(i in 1:n){
M_psi_nm[i,u_i[i]] = 1
M_Psi_nm[i,1:u_i[i]] = c(if(u_i[i]>1){Delta[1:(u_i[i]-1)]},
x[i]-Zeta[u_i[i]])
}
if(which.matrix[1]){M_psi_nm}else{M_Psi_nm}
}
#~
```

A.5 Simulating Mixture-Cure Data

In this section of the appendix, we discuss the simulation scheme of mixture-cure data. This simulation scheme is used for Paper 3 which focussed on mixture-cure analysis.

A.5.1 R Code

# Step 1: Packa	ges and Options
# Packagos	
# rackayes	
library(Matrix)	
library(tidyver	se)
library(surviva	1)
library(smcure)	
# Options - no	scientific noitation
options(scipen=	999)
# Supress autom	natic output of a function
quiet <- functi	on(x) {
<pre>sink(tempfile()</pre>)
on.exit(sink())	
invisible(force	$(\mathbf{x}))$
}	
#======================================	
#======	
# Step 2: FUNCI	IUW drawsim() to draw sample
drawSim <- func	tion(in.seed, sampSize, censor.ub, true.alpha0){
#~~~~~~	
# Simulation of	mixture-cure survival times
#	
# in.seed	= seed for random numbers
# sampSize	= size of survival sample to draw
# censor.ub	= upper bound for censoring distribution U(0,censor.ub)
#~~~~~~	NNNN
#~~~~~~~~	NNNN
# Step A: Setup	
# Seed, sample	size, basline
<pre>set.seed(in.see</pre>	d)
n=sampSize	
lambda = 1 # Sc	ale parameter
nu = 1 # Sh	ape parameter
#~~~~~	NNNNN
#~~~~~~	val
#~~~~~~ # Step B: Survi	
#~~~~~~ # Step B: Survi # Baseline line	ar predictor
#~~~~~ # Step B: Survi # Baseline line x1=rnorm (n,mea	ar predictor n=0,sd=1)
#~~~~~~ # Step B: Survi # Baseline line x1=rnorm (n,mea beta1=1;beta2=1	ar predictor n=0,sd=1) ;
<pre>#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~</pre>	ar predictor n=0,sd=1) ;
# # Step B: Survi # Baseline line x1=rnorm (n,mea beta1=1;beta2=1 xbeta=x1*beta1 #	ar predictor n=0,sd=1) ;
<pre># Step B: Survi # Baseline line x1=rnorm (n,mea beta1=1;beta2=1 xbeta=x1*beta1 #</pre>	ar predictor n=0,sd=1) ; ~~~~~
<pre># Step B: Survi # Baseline line x1=rnorm (n,mea beta1=1;beta2=1 xbeta=x1*beta1 # # # # # Step C: Under</pre>	ar predictor n=0,sd=1) ; ~~~~~ take time.event simulation
<pre># Step B: Survi # Baseline line x1=rnorm (n,mea beta1=1;beta2=1 xbeta=x1*beta1 # # # # Step C: Under # Uniform(0,1)</pre>	ar predictor n=0,sd=1) ; take time.event simulation for probability inverse transform

time.event=(-log(u)/(lambda*exp(xbeta)))^(1/nu)
#
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Step D: Undertake time.censor simulation
time.censor=runir(n=n, min=0, max=censor.ub)
<i>"</i>
#~~~~~~~~~~~
Step E: Susceptibility
Logistic linear predictor
w1=rnorm (n,mean=0,sd=1)
alpha0=true.alpha0;alpha1=1;
waipna=aipnaw + wi^aipnai
ui=runif(n. min=0. max=1)
<pre>true.susceptible=ifelse(ui>=true.pi,1,0)</pre>
#
#
Step F: Time and status
time =1felse(time.event<=time.censor & true.susceptible==1, time.event, time.censor)
#www.www.www.www.ww
#
Step G: Combine the data
All the data
ul.all-uata.llame(time-time, status-status, x1-A1, x2-X2, w1-W1, w2-W2, time event time consort true nitrue ni
ui=ui, true.susceptible=true.susceptible)
Modelling data
df = df.all %>%
<pre>select(time=time, status=status, x1=x1, x2=x2, w1=w1, w2=w2) %>%</pre>
arrange(time)
<i>"</i>
#~~~~~~~~~~~
Step H: Returned list
<pre>final=list(df.all=df.all, df=df)</pre>
return(final)
#~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
s #
#
<pre># Step 3: FUNCTION fitOnSim() to fit ML and MC models</pre>
<pre>fitOnSim = function(tag, simNum, in.seed, sampSize, censor.ub, true.alpha0){ #</pre>
######################################
#
<pre># tag = identifier for set of simulatoins</pre>
<pre># simNum = simulation number within tag</pre>
<pre># in.seed = seed for drawing simulations</pre>
<pre># sampSize = size of survival sample to draw</pre>
<pre># censor.ub = upper bound for censoring distribution U(0,censor.ub) "</pre>
#~~~~~~~~~~~~
#
Step A: Draw a simulation
<pre>sim1=drawSim(in.seed=in.seed, sampSize=sampSize, censor.ub=censor.ub, true.alpha0=true.alpha0)</pre>

```
#~
# Step B: Survival mixture cure model with smcure
mixc <- quiet(smcure(Surv(time,status)~x1,</pre>
cureform=~w1
data=sim1$df.
model="ph",
Var = F))
#~~~~~~
# Step C: Survival mixture cure model with smcure
fit1=smcml(formula=Surv(time.status)~x1.
formula.cure~w1.
data=sim1$df,
basis='uniform',
n.knots=c(3,1).
max.iter=2000,
range.quant=c(0.2,0.8))
#~~
# Step D: Collate results.
# Names
names.Alpha=t(t(paste0('Alpha', 1:length(fit1$coef$Alpha)-1)))
names.Beta =t(t(paste0('Beta' , 1:length(fit1$coef$Beta))))
names.Theta=t(t(paste0('Theta', 1:length(fit1$coef$Theta))))
names.ABT =rbind(names.Alpha, names.Beta, names.Theta)
names.ABT.se=paste0('se.',names.ABT)
names.AB =rbind(names.Alpha, names.Beta)
# Initial info
cr.df=sim1$df.all %>% filter(true.susceptible==1)
censorRate =c('all', 'Info', 'censorRate' , 1-sum(cr.df$status)/dim(cr.df)[1])
longRunSurvRate=c('all', 'Info', 'longRunSurvRate', sum(sim1$df.all$true.susceptible) /sampSize)
seed.used =c('all', 'Info', 'seedused' , in.seed)
time.ml =c('ml', 'time', 'time' , t2)
time.mc =c('mc', 'time', 'time', 'time', 't1)
iter =c('ml', 'Info', 'iter' , fit1$loglik$iter)
condnoAlpha =c('ml', 'Info', 'condNoAlpha' , fit1$condnoAlpha)
# Paramaters and se for ML
temp1=t(t(c(fit1$coef$Alpha, fit1$coef$Beta, fit1$coef$Theta)))
temp2=t(t(fit1$matricies$se.Eta_H))
parms.smcml=cbind('ml', 'Parm', names.ABT , temp1)
se.smcml =cbind('ml', 'SE', names.ABT.se, temp2)
# Paramaters for MC
temp3=t(t(c(mixc$b.mixc$beta))):rownames(temp3)=NULL
parms.smcure=cbind('mc', 'Parm', names.AB, temp3)
results=rbind(censorRate, longRunSurvRate, seed.used, time.ml, time.mc, iter, condnoAlpha,
parms.smcml, parms.smcure, se.smcml)
rownames(results)=NULL
colnames(results)=c('Model', 'Type', 'Parm', 'Val')
#~~~
# Step E: Output to csv
# Name of csv
                =str_replace_all(sampSize , "\\.", 'p')
name.sampSize
name.censor.ub =str_replace_all(censor.ub , "\\.", 'p')
name.true.alpha0 =str_replace_all(true.alpha0, "\\.", 'p')
```

<pre>name.true.alpha0 =str_replace_all(true.alpha0, "\\-", 'n')</pre>	
<pre>firstPart=paste0(tag,</pre>	
'_Reps_' ,name.sampSize,	
'_CensUB_' ,name.censor.ub,	
'_trueA0_' ,name.true.alpha0)	
#~~~~~~~~~~~	
#~~~~~~~~~~	
# Step F: Return results to R	
<pre>final=list(sim1=sim1, mixc=mixc, fit1=fit1, results=results)</pre>	
return(final)	
#~~~~~~~~~~~	
}	
#	

References

- Aalen, O. (1978). Nonparametric inference for a family of counting processes Annals of Statistics, 6(4), 701-726. https://www.jstor.org/stable/2958850
- ABS. (2020). 5206.0 Australian National Accounts: National Income, Expenditure and Product, Dec 2019 ABS. https://tinyurl.com/y36zk96z
- Alves, B. and Dias, J. (2015). Survival mixture models in behavioral scoring *Expert Systems* with Applications, 42(8), 3902–3910. https://doi.org/10.1016/j.eswa.2014.12.036
- Ambrose, B. and Capone, C. (2020). The Hazard Rates of First and Second Default *The Journal of Real Estate Finance and Economics*, 20(3), 275–293. https://doi.org/10.1023/A:1007837225924
- Anderson, R. (20074). The Credit Scoring Toolkit Oxford.
- Andersson, T., Dickman, P., Eloranta, S., Lambert, P. (2011). Estimating and Modelling Cure in Population-Based Cancer Studies Within the Framework of Flexible Parametric Survival Models *BMC Medical Research Methodology*, 11(96), 1-11. https://doi.org/10.1186/1471-2288-11-96
- Andersen, P. K. and Ørnulf, B. and Richard, D. G. and Niels, K. (1993). *Statistical models based on counting processes*. Springer New York.
- Apostolik, R., Donohue, C., Went, P. (2009). Foundations of Banking Risk, An Overview of Banking, Banking Risks and Risk-Based Banking Regulation *GARP*.
- APRA. (2020). Monthly Authorised Deposit-Taking Institution Statistics APRA. https://www.apra.gov.au

- Austin, P. (2012). Generating survival times to simulate cox proportional hazards models with time-varying covariates. *Statistics in Medicine*, 31(29), 3946-3958. https://doi.org/10.1002/sim.5452
- Austin, P., Lee, D., Fine, J. (2016). Introduction to the Analysis of Survival Data in the Presence of Competing Risks. *Circulation*, 133(6), 601-609. https://doi.org/10.1161/CIRCULATIONAHA.115.017719
- Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society*, 54(6), 627-635. https://doi.org/10.1057/palgrave.jors.2601545
- Banasik, J., Crook, J., Thomas, L. (1999). Not if but when will borrowers default. *Journal* of the Operational Research Society, 50(12), 1185-1190. https://doi.org/10.2307/3010627
- Bakoyannis, G. and Touloumi, G. (2011). Practical methods for competing risks data: A review *Statistical Methods in Medical Research*, 21(3), 257-272. https://doi.org/10.1177/0962280210394479
- Barnett, A. and Graves, N. (2008). Competing risks models and timedependent covariates *Critical Care (Biomed Central)*, 12(134), 1-3. https://ccforum.biomedcentral.com/articles/10.1186/cc6840
- Basu, S. and Tiwari, R. (2010). Breast cancer survival, competing risks and mixture cure model: a Bayesian analysis *Journal of the Royal Statistical Society*. *Series A (Statistics in Society)*, 173(2), 307-329. https://doi.org/10.1111/j.1467-985X.2009.00618.x
- Bellotti, T. and Crook, J. (2009). Credit scoring with macroeconomic variables using survival analysis. *Journal of the Operational Research Society*, 60(12), 1699-1707. https://doi.org/10.1057/jors.2008.130
- Bellotti, T., and Crook, J. (2013). Forecasting and stress testing credit card default using dynamic models. International Journal of Forecasting, 29(4), 563-574. https://doi.org/10.1016/j.ijforecast.2013.04.003

- Bellotti, T. and Crook, T. (2014). Retail credit stress testing using a discrete hazard model with macroeconomic factors *Journal of the Operational Research Society*, 65(3), 340-350. https://doi.org/10.1057/jors.2013.91
- Bender, R., Augustin, T., and Blettner, M. (2005). Generating survival times to simulate Cox proportional hazards models. *Statistics in Medicine*, 24(11), 1713-1723. https://doi.org/10.1002/sim.2059
- Berkson, Y. and Gage, R. (1952). Survival Curve for Cancer Patients Following Treatment. J. Amer. Statist. Assoc., 47(259), 501-515. https://www.jstor.org/stable/2281318
- Bernoulli, D. (1760). Essai d'une nouvelle analyse de la mortalité causée par la petite vérole, et des avantages de l'inoculation pour la prévenir *Mémoir de l'Academie Royale de Science*, 1-456.
- Beyersmann, J., Allignol, A., Schumacher, M. (2012). Competing Risks and Multistate Models with R *Springer*, New York, NY.
- Beyersmann, J., Latouche, A., Buchholz, A., Schumacher, M. (2009). Simulating competing risks data in survival analysis *Statistics in Medicine*, 28(6), 956-971. https://doi.org/10.1002/sim.3516
- BIS. (2005). An Explanatory Note on the Basel II IRB Risk Weight Functions *Bank for International Settlements*. https://www.bis.org/bcbs/irbriskweight.htm
- BIS. (2006). International Convergence of Capital Measurement and Capital Standards A Revised Framework Comprehensive Version *Bank for International Settlements*. Basel Framework
- Breslow, N. E. (1972). Contribution to the discussion of paper by D.R. Cox. *Journal of the Royal Statistical Society: Series B*, 34, 216–217 https://www.jstor.org/stable/2985181
- Cai, T. and Betensky, R. A. (2003). Hazard regression for interval-censored data with penalized spline. *Biometrics*, 59(3), 570-579. http://www.jstor.org/stable/3695433
- Cai, T., Hyndman, R., Wand, M. (2012). Mixed Model-Based Hazard Estimation *Journal of Computational and Graphical Statistics*, 11(4), 784-798. https://doi.org/10.1198/106186002862

- Cai, C., Zou, Y., Peng, Y., Zhang, J. (2012). smcure: Fit Semiparametric Mixture Cure Models *R package version 2.0*, https://CRAN.R-project.org/package=smcure
- Chan, R. H. and Ma, J. (2012). A multiplicative iterative algorithm for box-constrained penalized likelihood image restoration. *IEEE Trans. Image Processing*, 21, 3168-3181. https://doi.org/10.1109/TIP.2012.2188811
- Clapp, J., An, X., and Deng, Y. (2005). smcure: Unobserved Heterogeneity in Models of Competing Mortgage Termination. *Social Science Research Network*. https://ssrn.com/abstract=512624.
- Corbiere, F., Commenges, D., Taylor, J., Joly, P. (2009). A penalized likelihood approach for mixture cure models *Statistics in Medicine*, 28(3), 510–524. https://doi.org/10.1002/sim.3481
- Corbiere, F., Joly, P. (2007). A SAS macro for parametric and semiparametric mixture cure models *Computer Methods and Programs in Biomedicine*, 85, 173–180. https://doi.org/10.1016/j.cmpb.2006.10.008
- Coelho-Barros, E., Achcar, J., Mazucheli, J. (2015). Mixture and Non-mixture Cure Rate Model Considering the Burr XII Distribution. In: Steland A., Rafajłowicz E., Szajowski K. (eds) Stochastic Models, Statistics and Their Applications. Springer Proceedings in Mathematics & Statistics. https://doi.org/10.1007/978-3-319-13881-7_24
- Cox, D. (1972). Regression models and life tables. *Journal of the Royal Statistical Society, Series B*, 34(2), 187-220. https://doi.org/10.1111/j.2517-6161.1972.tb00899.x
- Cox, D. (1975). Partial likelihood. *Biometrika*, 62(2), 269-276. https://doi.org/10.1093/biomet/62.2.269
- Cox, D. R. and Oakes, D. (1984). *Analysis of Survival Data*. Chapman and Hall, New York, NY.
- Crouchy, M., Galai, D., Mark, R. (2006). The Essentials of Risk Management *McGraw Hill*, New York, NY.

- Crowley, J. and Hu, M. (1977). Covariance analysis of heart transplant survival data. *Journal of the American Statistical Association*, 72(357), 27-36. https://doi.org/10.1080/01621459.1977.10479903
- David, J. and Moeschberger M. (1978). The Theory of Competing Risks Griffin, London.
- DeBoor, C. and Daniel, J. W. (1974). Splines with nonnegative B-spline coefficients. *Mathematics of Computation*, 28(126), 565-568. https://www.jstor.org/stable/2005928
- Deshmukh, S. (2012). Multiple Decrement Models in Insurance Springer, India.
- Dempster, A., Laird, N., and Rubin, D. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1), 1-38. https://doi.org/10.1111/j.2517-6161.1977.tb01600.x
- Deng, Y., Quigley, J., Van Order, R. (2000) Mortgage termination heterogeneity and the exercise of mortgage options. Econometrica, 68(2), 275-307. https://doi.org/10.1111/1468-0262.00110
- Dirick, L., Claeskens, G., Baesens, B. (2015). An Akaike Information Criterion for Multiple Event Mixture Cure Models *European Journal of Operational Research*, 241(2), 449-457. https://doi.org/10.1016/j.ejor.2014.08.038
- Dirick, L., Claeskens, G., and Baesens, B. (2017). Time to default in credit scoring using survival analysis: a benchmark study. *Journal of the Operational Research Society*, 68(6), 252-665. https://doi.org/10.1057/s41274-016-0128-9
- Dirick, L. Bellotti, T., Claeskens, G., Baesens, B. (2016). Macro-economic factors in credit risk calculations: Including time-varying covariates in mixture cure models. *Journal of Business and Economic Statistics*, 37(1), 40-53. https://doi.org/10.1080/07350015.2016.1260471
- Djeundje, V. and Crook, J. (2018). Incorporating heterogeneity and macroeconomic variables into multi-state delinquency models for credit cards *European Journal of Operational Research*, 271(2), 697-709. https://doi.org/10.1016/j.ejor.2018.05.040

- Djeundje , V., Crook, J. (2009). Dynamic survival models with varying coefficients for credit risks *European Journal of Operational Research*, 275(1), 319-333. https://doi.org/10.1016/j.ejor.2018.11.029
- Elorant. S. Lambert, P., Andersson, T., Björkholm, M., Dickman, P. (2014). The Application of Cure Models in the Presence of Competing Risks A Tool for Improved Risk Communication in Population-Based Cancer Patient Survival *Epidemiology*, 25(5), 742-748. https://doi.org/10.1097/EDE.000000000000130
- Erlenmaier, U. (2011). Chapter 4 in The Basel 2 Risk Parameters Estimation, Validation, and Stress Testing (2nd edition) *Springer*, New York, NY.
- Farewell, V. (1982). The Use of Mixture Models for the Analysis of Survival Data With Long-Term Survivors. *Biometrics*, 38(4), 1041-1046. https://www.jstor.org/stable/2529885
- Farewell, V. (1986). Mixture models in survival analysis: are they worth the risk? *Canadian Journal of Statistics*, 14(3), 257–262. https://www.jstor.org/stable/3314804
- Financial Accounting Standards Board (2016). Financial Instruments—Credit Losses (Topic 326) No. 2016-13 Measurement of Credit Losses on Financial Instruments. FASB Accounting Standards
- Fine, P. and Gray, R. (1999). A Proportional Hazards Model for the Subdistribution of a Competing Risk. *Journal of the American Statistical Association*, 94(466), 496-509. https://doi.org/10.1080/01621459.1999.10474144
- Fisher, L., Lin, D. (1999). Time-Dependent Covariates In The Cox Proportional-Hazards Regression Model Annual Review of Public Health, 20(1), 145-157. https://doi.org/10.1146/annurev.publhealth.20.1.145
- Good, I. (1950). Probability and the Weighting of Evidence Charles Griffith, London.
- Gray, R. (1988). A Class of K-Sample Tests for Comparing the Cumulative Incidence of a Competing Risk *The Annals of Statistics*, 16(13), 1141-1154. https://www.jstor.org/stable/2241622

- Gu, Y., Sinha, D., Banerjee, S. (2011). Analysis of Cure Rate Survival Data Under Proportional Odds Model *Lifetime Data Anal.*, 17(1), 123–134. https://doi.org/10.1007/s10985-010-9171-z
- Hall, A. and Lundstedt, K. (2005). The Competing Risks Framework for Mortgages: Modeling the Interaction of Prepayment and Default *Risk Management Association Journal*, September. https://tinyurl.com/yxhptgqv
- Halley, E. (1693). An estimate of the degrees of the mortality of mankind, drawn from curious tables of the births and funerals at the city of Breslaw *Philosophical Transactions* of the Royal Society of London, 17 https://doi.org/10.1098/rstl.1693.0007
- Hayden, E., and Porath, D. (2011). Chapter 1 in The Basel 2 Risk Parameters Estimation, Validation, and Stress Testing (2nd edition) *Springer*, New York, NY.
- Heinze, G. and Dunkler, D. (2008). Avoiding infinite estimates of time-dependent effects in small-sample survival studies. *Statistics in Medicine*, 27, 6455 6469. https://doi.org/10.1002/sim.3418
- Honore, B. E. and Powell, J. L. (1994). Pairwise difference estimators of censored and truncated regression models. *Journal of Econometrics*, 64(1-2), 241-278. https://doi.org/10.1016/0304-4076(94)90065-5
- Hosmer, D., Lemeshow, S., and May, S. (2008). Applied Survival Analysis: Regression Modeling of Time to Event Data, (2nd ed.). Wiley-Interscience, Hoboken, New Jersey.
- Hua, T. and Xiang, L. (2013). Efficient Estimation for Semiparametric Cure Models with Interval-Censored Data *Journal of Multivariate Analysis*, 121, 139-151. https://doi.org/10.1016/j.jmva.2013.06.006
- IASB. (2014). International Financial Reporting Standard 9. https://www.ifrs.org/issuedstandards/list-of-standards/ifrs-9-financial-instruments/
- Im, J., Apley, D., Qi, C., Shan, X. (2012) A time-dependent proportional hazards survival model for credit risk analysis. *Journal of the Operational Research Society*, 63(3), 306-321. https://doi.org/10.1057/jors.2011.34

- Joly, P., Commenges, D., and Letenneur, L. (1998). A penalized likelihood approach for arbitrarily censored and truncated data: Application to age-specific incidence of dementia. *Biometrics*, 54(1), 185-194. https://pubmed.ncbi.nlm.nih.gov/9574965/
- Johansen, S. (1983). An Extension of Cox's Regression Model *International Statistical Review*, 51(2), 165-174. https://www.jstor.org/stable/1402746
- Jorion, P. (2006). Value at Risk: The New Benchmark for Managing Financial Risk (3rd ed.) *McGrath Hill*, New York.
- Kalbfleisch, J. D. and Prentice, R. L. (2002). *The Statistical Analysis of Failure Time Data*. John Wiley and Sons, New York, NY.
- Kaplan. E. and Meier, P. (1958). Nonparametric estimation from incomplete observations. *The Journal of the American Statistical Association*, 53(282), 457-481. https://doi.org/10.1080/01621459.1958.10501452
- Karush, W. (1939). Minima of functions of several variables with inequalities as side constraints. [Master's thesis, M.Sc. Dissertation. Dept. of Mathematics], Univ. of Chicago.
- Klein, J., and Moeschberger, M. (2003). Survival Analysis Techniques for Censored and Truncated Data (2nd ed.) *Springer*, New York, NY.
- Kneib, T, and Fahrmeir, L. (2004). A mixed model approach for structured hazard regression *Collaborative Research Center 386*, *Discussion*, 400. https://doi.org/10.5282/ubm/epub.1770
- Kuhn, H. and Tucker (1951). Nonlinear programming. *Proceedings of 2nd Berkeley Symposium. Berkeley*, pp481-492.
- Lam, J. (2003). Enterprise Risk management: From Incentives to Controls Wiley Finance.
- Lessmann, S., Baesens, B., Seowd, H., and Thomas, L. (2015). Benchmarking state-of-theart classification algorithms for credit scoring: An update of research. *The Journal of the Operational Research Society*, 247(1), 124-136. https://doi.org/10.1016/j.ejor.2015.05.030
- Li, Y., Tiwari, R., Guha, S. (2007). Mixture Cure Survival Models with Dependent Censoring *Journal*,69(3), 285-306. https://doi.org/10.1111/j.1467-9868.2007.00589.x

- Liu, H. and Shen, Y. (2009). A Semiparametric Regression Cure Model for Interval-Censored Data *Journal of the American Statistical Association*, 104(487), 1168-1178. https://doi.org/10.1198/jasa.2009.tm07494
- Lu, W. (2008). Maximum likelihood estimation in the proportional hazards cure model. *Ann Inst Stat Math*, 60, 545-574. https://doi.org/10.1007/s10463-007-0120-x
- Luptakova, I. and Bilikova, M. (2014). Actuarial modeling of life insurance using decrement models *Journal of Applied Mathematics*, *Statistics and Informatics*, 10(1), 81-91. https://doi.org/10.2478/jamsi-2014-0008
- Luenberger, D. and Ye, Y. (2008). *Linear and Nonlinear Programming, 3 ed.* Springer, New York, NY.
- Ma, J. (2010). Positively constrained multiplicative iterative algorithm for maximum penalized likelihood tomographic reconstruction. *IEEE Transactions on Nuclear Science*, 57(1), 181-192. https://doi.org/10.1109/TNS.2009.2034462
- Ma, J., Heritier, S., and Lô, S. (2014). On the maximum penalized likelihood approach for proportional hazard models with right censored survival data. *Computational Statistics and Data Analysis*, 74, 142-156. https://doi.org/10.1016/j.csda.2014.01.005
- Man, R. (2014). Survival analysis in credit scoring A framework for PD estimation *Masters Thesis*, University of Twente, The Netherlands. https://tinyurl.com/y3ntqlph
- Mantel, N. (1966). Evaluation of survival data and two new rank order statistics arising in its consideration *Cancer Chemotherapy Reports*, 50, 163-170.
- McNeil, A., Frey, R., Embrechts, P. (2005). Quantitative Risk Management *Princeton University Press.*
- Mohammad, K., Hirose, Y., Surya, B., Yao, Y. (2019). Efficient Estimation For The Cox Proportional Hazards Cure Model *Working Paper*.
- Moore, T. J., Sadler, B. M. and Kozick, R. J. (2008). Maximum-likelihood estimation, the Cramer-Rao bound, and the method of scoring with parameter constraints. *IEEE Transactions of Signal Process*, 56, 895-908. https://ieeexplore.ieee.org/document/4451293

- Murphy, S. (1994) Consistency in a proportional hazards model incorporating a random effect. *Annals of Statistics*, 22(2), 721–731. https://www.jstor.org/stable/2242287
- Murphy, S. (1995) Asymptotic theory for the frailty model. *Annals of Statistics*, 23(1), 182-198. https://www.jstor.org/stable/2242406
- Narain, B. (1992). Survival analysis and the credit granting decision, in Credit Scoring and Credit Control, L. C. Thomas, J. N. Crook, D. B. Edelman, eds. *Oxford University Press*.
- Nelson, W. (1969). Hazard plotting for incomplete failure data *Journal of Quality Technology*, 1, 27-52. https://doi.org/10.1080/00224065.1969.11980344
- Nelson, W. (1972). Theory and applications of hazard plotting for censored failure data *Technometrics*, 14(4), 945-966. https://www.jstor.org/stable/1267144
- OED. (2020). Oxford English Dictionary Oxford English Dictionary. http://www.oed.com
- Patilea, V., VanKeilegom, I. (2017). A General Approach for Cure Models in Survival Analysis. arXiv: Statistics Theory, 1-33. https://arxiv.org/pdf/1701.03769
- Peto, R., and Peto, J. (1972). Asymptotically Efficient Rank Invariant Test Procedures Journal of the Royal Statistical Society. Series A (General), 135(2), 185-207. https://doi.org/10.2307/2344317
- Peng, Y. and Dear, K. (2000). A Nonparametric Mixture Model for Cure Rate Estimation. *Biometrics*, 56(1), 237-243. https://www.jstor.org/stable/2677127
- Pintilie, M. (2006). Competing Risks a Practical Perspective John Wiley and Sons, West Sussex, England.
- Prentice, R. L., Kalbfleisch, J. D., Peterson, A. V., Flournoy, N., Farewell, V. T. and Breslow,
 N. E. (1978). The analysis of fa7ilure times in the presence of competing risks. *Biometrics*, 34(4), 541-554. https://doi.org/10.2307/2530374
- Putter, H., Fiocco, M. and Geskus, R. B. (2007). Tutorial in biostatistics: competing risks and multi-state models. *Statistics in Medicine*, 26(11), 2389-2430. https://doi.org/10.1002/sim.2712

- R Core Team. (2020). *R* 4.0.0: A Language and Environment for Statistical Computing. *R* Foundation for Statistical Computing. http://www.R-project.org
- Ramsay, J. O. (1988). Monotone regression splines in action. *Statistical Science*, 3(4), 425-441. https://doi.org/10.1214/ss/1177012761
- Ren, J., and Zhou, M. (2011). Full likelihood inferences in the Cox model: an empirical likelihood approach *Annals of the Institute of Statistical Mathematics*, 63(5), 1005-1018. https://doi.org/10.1007/s10463-010-0272-y
- Rodriguez, G. (2005). Non-Parametric Estimation in Survival Models *Princeton Lecture Notes*, A(B4), CC-DD6. https://data.princeton.edu/pop509/NonParametricSurvival.pdf
- Royston, P. (2011). Estimating a smooth baseline hazard function for the Cox model *Working Paper*. https://tinyurl.com/y6bkjx8g
- Rutkowski, M., Tarca, S. (2015). Regulatory Capital Modelling for Credit Risk International Journal of Theoretical and Applied Finance, 18(05). https://doi.org/10.1142/S021902491550034X
- SAS Institute Inc. (2020). SAS Software, Version 9.4. http://www.sas.com
- Schoenfeld, D. (Yr1981). The asymptotic properties of nonparametric tets for comparing survival distributions *Biometrika*, 68(1), 316-319. https://doi.org/10.2307/2335833
- Siddiqi, N. (2005). *Credit Risk Scorecards: Developing and Implementing Intelligent Credit Scoring*. Wiley and SAS Business Series.
- Stepanova, M. and Thomas, L. (2002). Survival analysis methods for personal loan data. *Operations Research*, 50(2), 277-289. https://doi.org/10.1287/opre.50.2.277.426
- Sterne, J., White, I., Carlin, J., Spratt, M., Royston, P., Kenward, M., Wood, A., Carpenter, J. (2009). Multiple imputation for missing data in epidemiological and clinical research: potential and pitfalls *British Medical Journal*,338. https://doi.org/10.1136/bmj.b2393
- Sueyoshi, G. (1992). Semiparametric proportional hazards estimation of competing risks models with time-varying covariates *Journal of Econometrics*, 51(1-2), 25-58. https://doi.org/10.1016/0304-4076(92)90028-P

- Sy, J. and Taylor, J. (2000). Estimation in a Cox Proportional Hazards Cure Model. *Biometrics*, 56(1), 227-236. https://doi.org/10.1111/j.0006-341X.2000.00227.x
- Thackham, M., and Ma, J. (2019) Exposure at default without conversion factors evidence from Global Credit Data for large corporate revolving facilities *J. R. Statist. Soc. A*, 182(4), 1267-1286. https://doi.org/10.1111/rssa.12418
- Thackham, M. and Ma, J. (2020). On maximum likelihood estimation of the semi-parametric Cox model with time-varying covariates. *Journal of Applied Statistics*, 47(9), 1511-1528. https://doi.org/10.1080/02664763.2019.1681946
- Thackham, M. and Ma, J. (2020). On maximum likelihood estimation of competing risks using the cause-specific semi-parametric Cox model with time-varying covariates An application to credit risk *Journal of the Operational Research Society*, xx(y), aaa-bbb. https://doi.org/10.1080/01605682.2020.1800418
- Therneau, T. M. (2019). *A Package for Survival Analysis in R*. (R package version 3.1-6). https://CRAN.R-project.org/package=survival
- Therneau, T., Crowson, C., and Atkinson, E. (2015). Using time dependent covariates and time dependent coefficients in the Cox model. R CRAN Vignette. https://cran.rproject.org/web/packages/survival/vignettes/timedep.pdf
- Tong, T., Mues, C., and Thomas, L. (2012). Mixture cure models in credit scoring: If and when borrowers default. *European Journal of Operational Research*, 218(1). 132-139. https://doi.org/10.1016/j.ejor.2011.10.007
- Tong, T., Mues, C., Thomas, L. (2013). A zero-adjusted gamma model for mortgage loan loss given default. *International Journal of Forecasting*, 29(4), 548-562. https://doi.org/10.1016/j.ijforecast.2013.03.003
- vanHouwelingen, H. (2000). Validation, calibration, revision and combination of prognostic survival models *Journal*, 19(24), 3401-3415. https://doi.org/10.1002/1097-0258(20001230)19:24<3401::AID-SIM554>3.0.CO;2-2
- Vasicek, O. (2002). Loan portfolio value Cutting Edge, 160-162. Risk.net

- Wong, W. H. and Severini, T. A. (1991). On maximum likelihood estimation in infinite dimensional parameter spaces. Ann. Statist., 19(2), 603-632. https://doi.org/10.1214/aos/1176348113
- Wong, G. Y. C., Osborne, M. P., Diao, Q., Yu, Q. (2017). Piecewise Cox models with rightcensored data. Communications in Statistics - Simulation and Computation, 46()10), 7894-7908. https://doi.org/10.1080/03610918.2016.1255968
- Wycinka, E. (2019) Competing Risk Models of Default in the Presence of Early Repayments. Econometrica, 23(2), 99–120. https://doi.org/10.15611/eada.2019.2.07
- Xu, J. and Scott-Long, J. (2005). Confidence intervals for predicted outcomes in regression models for categorical outcomes. *The Stata Journal*, 5(4), 537–559. https://doi.org/10.1177/1536867X0500500405
- Xu, J., Ma, J., Connors, M. H. and Brodaty, H. (2018). Proportional hazard model estimation under dependent censoring using copulas and penalized likelihood. *Statistics in Medicine*, 37(14), 2238-2251. https://doi.org/10.1002/sim.7651
- Yin, G., and Ibrahim, J. (2005). Cure rate models a unified approach *The Canadian Journal of Statistics*, 33(4), 559-570. https://doi.org/10.1002/cjs.5550330407
- Yeh, A., Twaddle, J., Frith, M. (2005). Reserve Bank Bulletin Reserve Bank of New Zealand, 68(3), 1-12. https://www.rbnz.govt.nz/research-and-publications/reserve-bankbulletin/2005/rbb2005-68-03-01
- Yu, Y. and Ruppert, D. (2002). Penalized spline estimation for partially linear single index models. J. Amer. Statist. Assoc., 97(460), 1042-1054. https://doi.org/10.1198/016214502388618861
- Yu, B., and Tiwari, R. (2007) Application of EM Algorithm to Mixture Cure Model for Grouped Relative Survival Data *Journal of Data Science*, 23(11), 1733-1747. https://doi.org/10.1002/sim.1774
- Zangwill, W. (1969). *Nonlinear Programming: A Unified Approach*. Prentice-Hall, New Jersey.

- Zhang Z., Reinikainen J., Adeleke K.A., Pieterse M.E., Groothuis-Oudshoorn C.G. (2018). *Time-varying covariates and coefficients in Cox regression models*. *Annals of Translational Medicine*, 6(7), 121. https://doi.org/10.21037/atm.2018.02.12
- Zheng, D. and Lin, D. (2007). Maximum likelihood estimation in semiparametric regression models with censored data. *Journal of the Royal Statistical Society B*, 69, 507-564. https://doi.org/10.1111/j.1369-7412.2007.00606.x