

Makro, mikro och maskininlärningsmetoder

Mathias Lindholm

Stockholms universitet

SAf, Maj, 2022

Merparten av det som presenteras bygger på samarbeten med

- ▶ Łukasz Delong
- ▶ Filip Lindskog
- ▶ Richard Verrall
- ▶ Felix Wahl
- ▶ Mario Wüthrich
- ▶ Henning Zakrisson

Mål för idag

Diskutera reservsättning makro → mikro...

...mha traditionella regressionsmodeller

...mha maskininlärningsmetoder

Disposition

- ▶ Allmänt om makro- och mikromodeller
- ▶ Chain-ladder modeller: antaganden och regression
- ▶ Double chain-ladder och skadenatal
- ▶ Mikro-modeller

Allmänt om makro- och mikromodeller

Makromodeller:

- ▶ utgår från aggregerad data
- ▶ lätta att använda (ej beräkningstunga)
- ▶ ofta robusta
- ▶ ofta väntevärdesriktiga (givet korrekta modellantaganden)
- ▶ makrodata in, makrodata ut

Mikromodeller:

- ▶ utgår direkt från kontraktsdata
- ▶ kräver ofta mer (explicita) antaganden
- ▶ kan ge högre precision, typiskt ej väntevärdesriktiga (förhoppningsvis konsistenta)
- ▶ kräver mer av användaren
- ▶ mikrodata in, (som mest) mikrodata ut

Allmänt om makro- och mikromodeller

Klassisk regression:

- ▶ Ofta möjligt att hantera modeller (delvis) analytiskt
- ▶ Inte för många parametrar i förhållande till data...
...går att hantera mha regularisering
- ▶ Kräver specificering av funktionella former
- ▶ Tolkningsbara

Maskininlärning:

- ▶ ...är väsentligen regression (supervised learning)
- ▶ “Slipper” definiera funktionella former (explicit)
- ▶ Kräver “mycket” data, ofta beräkningstungt
- ▶ Ofta svåra att tolka

Utsikter för att använda för makromodeller!?

Är möjligt, givet visst knep och knåp!

Chain-ladder modeller

Låt

- ▶ X_{ij} beteckna inkrementella betalningar från skadeår $i = 1, \dots, m$ betalda under utvecklingsår $j = 0, \dots, m - 1$
- ▶ C_{ij} beteckna motsvarande kumulativa betalningar
- ▶ R_i beteckna utestående betalningar för skadeår i

Dvs. vid slutet av skadeår m har vi att

$$\begin{aligned} R_i &:= \sum_{j=m-i+1}^{m-1} X_{ij} \\ &= C_{i,m-1} - C_{i,m-i} \end{aligned}$$

Chain-ladder modeller

- ▶ Låt \mathcal{F}_0 beteckna informationen som är känd t.o.m. slutet av skadeår m
- ▶ Antag att vi har en modell definierad itermer av en parametervektor θ
- ▶ Den *teoretiska* prediktorn ges då av

$$h_i(\theta; \mathcal{F}_0) := E[R_i | \mathcal{F}_0]$$

- ▶ Den *beräkningsbara* prediktorn ges då av

$$\hat{R}_i := h_i(\hat{\theta}; \mathcal{F}_0)$$

Chain-ladder modeller

- ▶ Chain-ladder “tekniken” motsvarar en specifik beräkningsbar prediktor
- ▶ Olika *modeller* kan producera samma beräkningsbara prediktor
 - ▶ Mack’s fördelningsfria chain-ladder modell (CLM)
 - ▶ Den överspridda Poisson chain-ladder modellen (ODP-CL)

Chain-ladder modeller

CLM:

- ▶ $E[C_{i,j} | C_{i,j-1}] = f_j C_{i,j-1}$
- ▶ $\text{Var}(C_{i,j} | C_{i,j-1}) = \sigma_j^2 C_{i,j-1}$ (behövs ej för prediktion)
- ▶ alla skadeår antas vara oberoende

ODP-CL:

- ▶ alla $X_{i,j}$ antas vara oberoende $\text{ODP}(\alpha_i \beta_j, \phi)$
 $\Rightarrow E[X_{i,j}] = \alpha_i \beta_j$ (med $\text{Var}(X_{i,j}) = \phi E[X_{i,j}]$)

$\Rightarrow \hat{R}^{CLM} = \hat{R}^{ODP}$ men $h^{CLM}(\boldsymbol{\theta}^{CL}; \mathcal{F}_0) \neq h^{ODP}(\boldsymbol{\theta}^{ODP}; \mathcal{F}_0)$

Chain-ladder modeller

För båda modeller är det möjligt att beräkna

- ▶ processvarians, dvs. $\text{Var}(R \mid \mathcal{F}_0)(\boldsymbol{\theta})$
- ▶ skattningsfel, dvs. $E[(h(\hat{\boldsymbol{\theta}}^*; \mathcal{F}_0) - h(\boldsymbol{\theta}; \mathcal{F}_0))^2; \mathcal{F}_0](\boldsymbol{\theta})$

så vi kan därmed beräkna MSEP

$$\widehat{\text{MSEP}}(R, \hat{R}; \mathcal{F}_0) := E[(R - h(\hat{\boldsymbol{\theta}}^*; \mathcal{F}_0))^2 \mid \mathcal{F}_0],$$

där $\hat{\boldsymbol{\theta}}^*$ motsvarar tex. oberoende kopior av $\hat{\boldsymbol{\theta}} \notin \mathcal{F}_0$ eller bootstrappa (se tex. Buchwalder et al. (2006), Lindholm, Lindskog & Wahl (2020))

Chain-ladder modeller: regression

- ▶ Med ODP-CL är vi klara (Poisson regression)
- ▶ För CLM kan vi notera att, givet $C_{i,j-1}$ följer att

$$C_{i,j} = C_{i,j-1}f_j + \sigma_j\sqrt{C_{i,j-1}}\epsilon_{i,j},$$

där $\epsilon_{i,j}$ är sådan att $E[\epsilon_{i,j}] = 0$, $\text{Var}(\epsilon_{i,j}) = 1\dots$

...vilket vi kan betrakta som en följd av generella linjära modeller

Vi kommer att börja med att fokusera på ODP-CL

Klassisk regression ODP-CL:

$$E[X_{i,j}] = \alpha_i \beta_j = \exp\{\tilde{\alpha}_i + \tilde{\beta}_j\}$$

Maskininlärning ODP-CL:

$$E[X_{i,j}] = \exp\{f(i,j; \boldsymbol{\theta})\},$$

där $f(i,j; \boldsymbol{\theta})$ är en funktion som ges (implicit) av ditt val av maskininlärningsmetod...

...där $\boldsymbol{\theta}$ typiskt är väldigt högdimensionell

Kort om maskininlärningsmetoder

Vi kommer att använda

- ▶ trädbaserade metoder - gradient boosting machines (GBM)

$$f(\mathbf{c}; \boldsymbol{\theta}_m) = f(\mathbf{c}; \boldsymbol{\theta}_{m-1}) + \sum_{j=1}^{2^d} \delta_{j,m} \mathbf{1}_{\{\mathbf{c} \in \mathcal{R}_{j,m}\}}$$

där m motsvarar antal träd, och d är djupet i trädet

- ▶ artificiella neuronnätverk (NN)

$$f(\mathbf{c}; \boldsymbol{\theta}_m) = g_m(\mathbf{c}) := u_m(\mathbf{a}_m + \mathbf{B}_m \mathbf{g}_{m-1}(\mathbf{c})), \quad \mathbf{g}_0(\mathbf{c}) := \mathbf{u}_0(\mathbf{c})$$

där m motsvarar antal lager i NN och dimensionerna på vektorer/matriser ger antal neuroner per lager

Generellt problem: överparametrisering

Lösning: “early stopping”

Kort om maskininlärningsmetoder: Early stopping

Grundidén bakom early stopping är följande:

- ▶ Oavsett modell görs kalibreringen iterativt
- ▶ Kalibreringen utvärderas mot en specifik “loss” eller förlustfunktion (tänk negativ log-likelihood)

Vi kommer att använda deviansresidualer (väsentligen MLE)

- ▶ Dela upp data i två delar: träningsdata och valideringsdata
- ▶ Givet startvärde θ_0 , iteration m innebär att
 - (a) ta fram θ_m baserat på träningsdata och θ_{m-1}
 - (b) utvärdera θ_m med förlustfunktionen på valideringsdataoch iterera så länge förlustfunktionen i (b) förbättras

Kort om maskininlärningsmetoder: Early stopping

...OK, men hur funkar detta för triangeldata!?

Förslag från Gabrielli et al. (2019):

- ▶ **OBS:** typiskt sett har vi många kontrakt även om vi inte nödvändigtvis har många skadeår / utv.år
- ▶ Dela upp kontrakten i två delar: en träningsdel och en valideringsdel
- ▶ Skapa utgående från dessa en träningstriangel och en valideringstriangel

OBS: behöver göras med viss försiktighet!

Enklast: dela upp kontrakten i två lika stora delar med ungefär lika många skador

Kort om maskininlärningsmetoder: boosting

Några ord om boosting:

- ▶ I Gabrielli et al. (2019) introduceras en “boostad” (eller “blended”) NN model för ODP-CL
- ▶ Detta innebär att de
 - (a) anpassar en vanlig ODP-CL
 - (b) använder väntevärdet från (a) som offset till en Poisson NN-modelldär (b) kan betraktas som att du anpassar en NN-modell till residualen från en ODP-CL
- ▶ För GBM:er så görs motsvarande i varje modelliteration med hjälp av träd (men utgående från gradienter)

Numeriskt exempel

Simulerar 6 st LoB:ar med en NN-modell anpassad till Schweizisk försäkringsdata, se Gabrielli & Wüthrich (2018)

Vi börjar med den NN-boostade ODP-CL från Gabrielli et al. (2019)

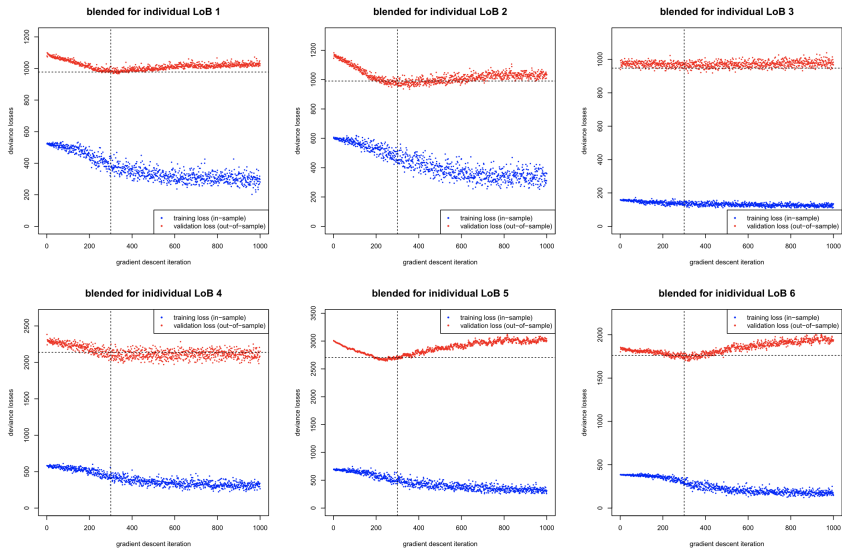


Figure 2. Out-of-sample validation analysis for over-fitting with neurons $(q_1, q_2, q_3) = (20, 15, 10)$ and dropout rates of 10% in all hidden layers: blue dots show in-sample model training and red dots show out-of-sample validation.

Från Gabrielli et al. (2019)

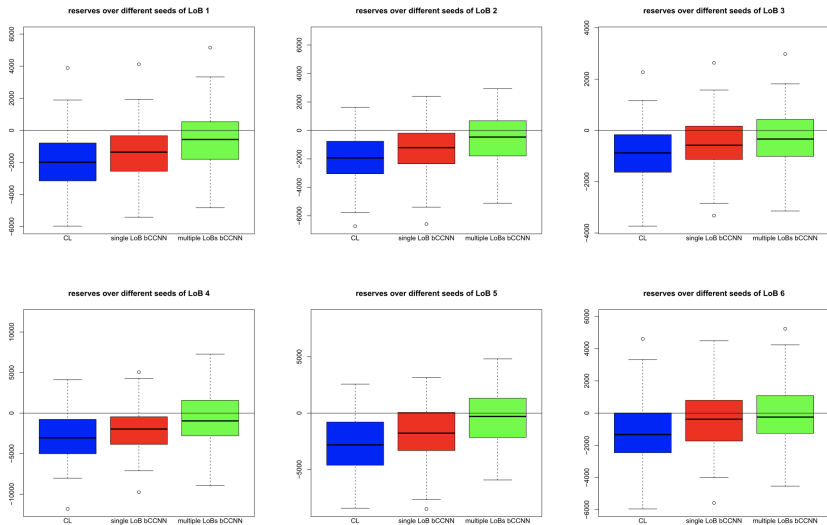


Figure 3. Comparison of the CL biases $R_m^{\text{CL}} - R_m^{\text{true}}$ (blue color), the single LoB bCCNN biases $R_m^{\text{LoB}} - R_m^{\text{true}}$ (red color) and the multiple LoBs bCCNN biases $R_m^+ - R_m^{\text{true}}$ (green color) over 100 simulations with different seeds (we also refer to Figure 1 in the appendix).

Från Gabrielli et al. (2019)

DCL-modeller

- ▶ ODP-CL är trots allt CL – kan t.ex. inte få fram RBNS / IBNR
- ▶ **Alternativ:** DCL-modeller, se Verrall et al. (2010), Miranda et al. (2012), baserat på ett Poissonantagande
- ▶ **Idé:** utgå från att det är *antal* skador som driver *betalningarna*
- ▶ I Wahl et al. (2019) utvidgas dessa modeller till att hantera multipla betalningar per skada – CRM
 - ▶ CRM är konstruktivt motiverad och bygger på Poissonprocesser i diskret tid
 - ▶ Producerar både RBNS- och IBNR-reserver
 - ▶ Kan ses som ett mellanting mellan makro/mikro

DCL-modeller: CRM

- ▶ Kortfattat CRM:
 - ▶ $N_{i,j} \sim \text{ODP}(\nu_{i,j}, \phi)$, alla $N_{i,j}$ ober.
 - ▶ $X_{i,j,k} \mid \mathcal{N}_0 \sim \text{ODP}(N_{i,j}\psi_{i,j,k}, \varphi)$, alla $X_{i,j,k}$ ober., **där k motsvarar betalningsfördröjning efter rapportering j**
 - ▶ Dvs. $N_{i,j}$ fungerar som exponering (och inkluderas som offset)
- ▶ Analogi “maskininlärning”:
 - ▶ $N_{i,j} \sim \text{ODP}(f_N(i, j; \theta_N), \phi)$, alla $N_{i,j}$ ober.
 - ▶ $X_{i,j,k} \mid \mathcal{N}_0 \sim \text{ODP}(N_{i,j}f_X(i, j, k; \theta_X), \varphi)$, alla $X_{i,j,k}$ ober.

Diverse naturliga variationer av ovan diskuteras i Lindholm, Verrall, Wahl & Zakrisson (2020)

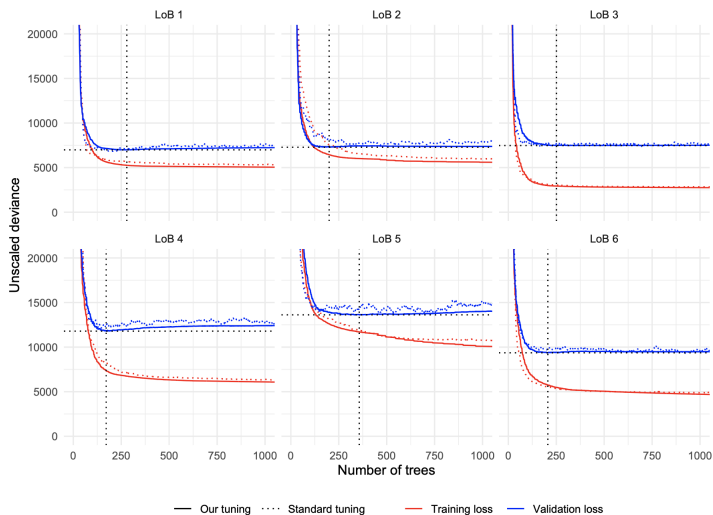


Figure 7: Out-of-sample validation analysis for over-fitting (determining the number of trees) in the GBM for Model 2. The dotted vertical and horizontal lines show the minimum of the validation loss and the number of trees at which it is obtained for our tuning.

Från Lindholm, Verrall, Wahl & Zakrisson (2020)

Table 4: True reserve compared to benchmark models and the GBMs and NNs. Relative biases of the reserve predictions in the parentheses.

	LoB 1	LoB 2	LoB 3	LoB 4	LoB 5	LoB 6
True reserves	39,689	37,037	16,878	71,630	72,548	31,117
CL reserves	38,569 (-2.82)	35,460 (-4.26)	15,692 (-7.02)	67,574 (-5.66)	70,166 (-3.28)	29,409 (-5.49)
CRM reserves	32,485 (-18.15)	29,901 (-19.27)	13,040 (-22.74)	55,782 (-22.12)	59,390 (-18.14)	24,403 (-21.58)
GCRM reserves	38,293 (-3.52)	35,117 (-5.18)	15,448 (-8.47)	66,961 (-6.52)	69,397 (-4.34)	29,104 (-6.47)
GRWNN	39,233 (-1.15)	35,899 (-3.07)	15,815 (-6.30)	70,219 (-1.97)	70,936 (-2.22)	30,671 (-1.43)
GBM (Model 2)	39,697 (0.02)	37,253 (0.58)	16,508 (-2.19)	72,679 (1.46)	71,828 (-0.99)	31,941 (2.65)
GBM (Model 2) without inflation	38,324 (-3.44)	37,053 (0.04)	16,327 (-3.26)	73,386 (2.45)	70,486 (-2.84)	32,100 (3.16)
GBM (Model 3)	40,114 (1.07)	35,729 (-3.53)	15,761 (-6.62)	69,448 (-3.05)	72,418 (-0.18)	30,061 (-3.39)
NN (Model 2)	41,587 (4.78)	37,587 (1.48)	15,680 (-7.10)	71,155 (-0.66)	71,309 (-1.71)	28,984 (-6.86)
NN (Model 3)	39,757 (0.17)	38,719 (4.54)	16,245 (-3.75)	70,916 (-1.00)	74,600 (2.83)	28,943 (-6.99)

Från Lindholm, Verrall, Wahl & Zakrisson (2020)

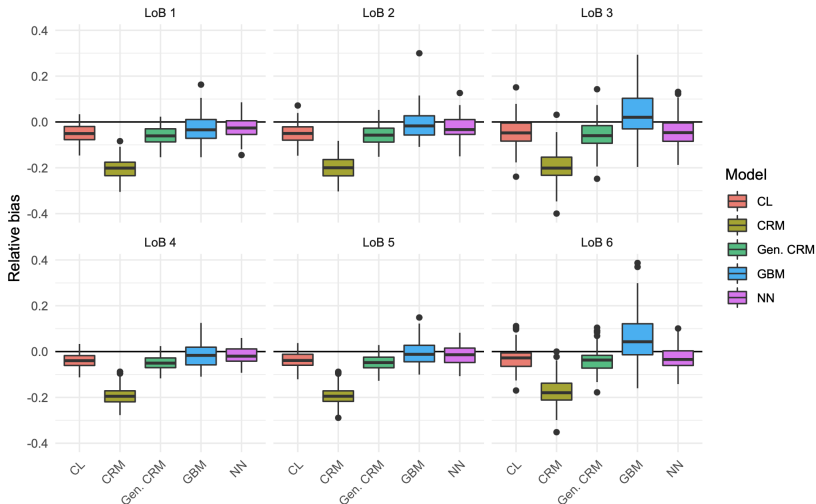


Figure 9: Boxplots of relative biases from the 100 simulations.

Från Lindholm, Verrall, Wahl & Zakrisson (2020)

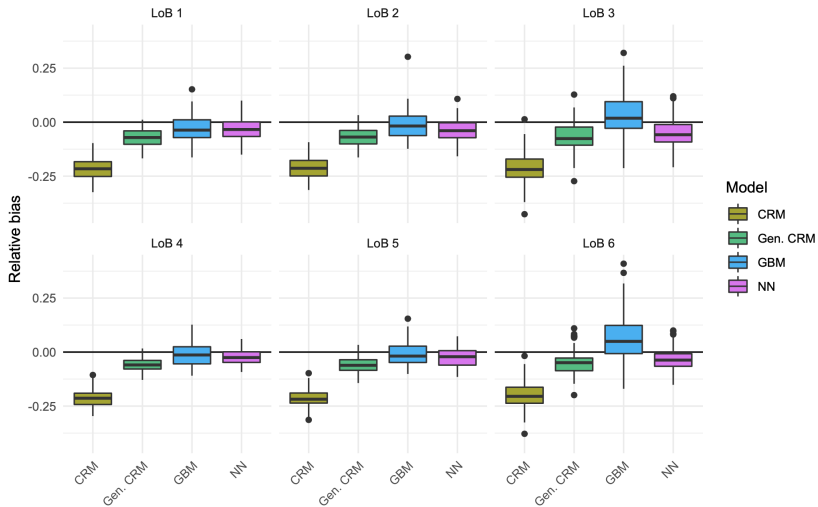


Figure 10: Boxplots of RBNS relative biases from the 100 simulations.

Från Lindholm, Verrall, Wahl & Zakrisson (2020)

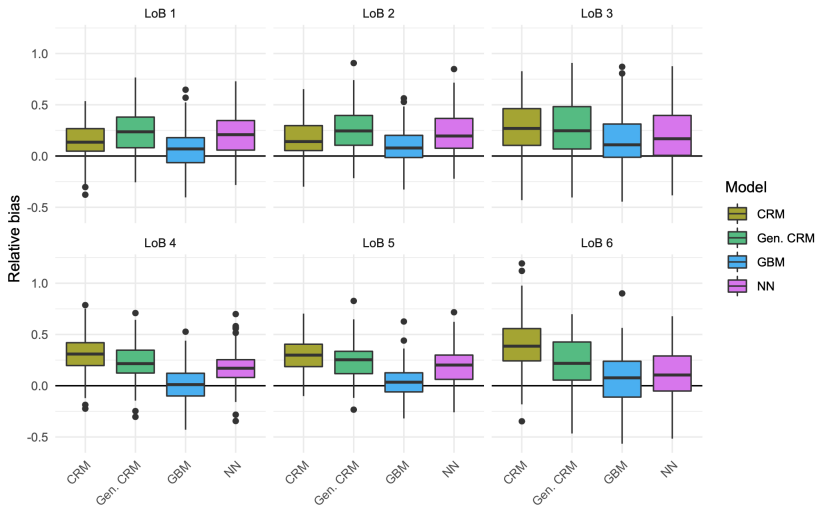


Figure 11: Boxplots of IBNR relative biases from the 100 simulations.

Från Lindholm, Verrall, Wahl & Zakrisson (2020)

Mikromodeller!?

- ▶ I någon mening “enklare” än makromodeller (närmre regression)
- ▶ Naturligt att fortsätta förfining i enlighet med CRM...
...(märkta) Poissonprocesser i kontinuerlig tid, se Norberg (1993), Antonio & Plat (2014)
- ▶ Ett exempel med NN ges i Delong et al. (2022)
- ▶ Tenderar att bli beräkningstungt!

Andra alternativ

- ▶ Makrostruktur på kontraktsnivå med kovariater
- ▶ Exempel Wüthrich (2018):

$$C_{i,j}(\mathbf{x}) = C_{i,j-1}(\mathbf{x})f(\mathbf{x}) + \sigma_j \sqrt{C_{i,j-1}(\mathbf{x})} \epsilon_{i,j}$$

- ▶ Mer konstruktiv modellering på aggregerad nivå!

Ex. ta med återöppning / stängning av skador, se Lindholm & Zakrisson (2022)

Tack!

Frågor?

Referenser I

- Antonio, K. & Plat, R. (2014), 'Micro-level stochastic loss reserving for general insurance', *Scandinavian Actuarial Journal* **2014**(7), 649–669.
- Buchwalder, M., Bühlmann, H., Merz, M. & Wüthrich, M. V. (2006), 'The mean square error of prediction in the chain ladder reserving method (mack and murphy revisited)', *ASTIN Bulletin: The Journal of the IAA* **36**(2), 521–542.
- Delong, Ł., Lindholm, M. & Wüthrich, M. V. (2022), 'Collective reserving using individual claims data', *Scandinavian Actuarial Journal* **2022**(1), 1–28.
- Gabrielli, A., Richman, R. & Wüthrich, M. V. (2019), 'Neural network embedding of the over-dispersed poisson reserving model', *Scandinavian Actuarial Journal* pp. 1–29.
- Gabrielli, A. & Wüthrich, M. V. (2018), 'An individual claims history simulation machine', *Risks* **6**(2), 29.

Referenser II

- Lindholm, M., Lindskog, F. & Wahl, F. (2020), 'Estimation of conditional mean squared error of prediction for claims reserving', *Annals of Actuarial Science* pp. 1–36.
- Lindholm, M., Verrall, R., Wahl, F. & Zakrisson, H. (2020), Machine learning, regression models, and prediction of claims reserves, in 'Casualty Actuarial Society E-Forum, Summer 2020'.
- Lindholm, M. & Zakrisson, H. (2022), 'A collective reserving model with claim openness', *ASTIN Bulletin: The Journal of the IAA* **52**(1), 1–27.
- Miranda, M. D. M., Nielsen, J. P. & Verrall, R. (2012), 'Double chain ladder', *ASTIN Bulletin: The Journal of the IAA* **42**(1), 59–76.
- Norberg, R. (1993), 'Prediction of outstanding liabilities in non-life insurance 1', *ASTIN Bulletin: The Journal of the IAA* **23**(1), 95–115.

Referenser III

- Verrall, R., Nielsen, J. P. & Jessen, A. H. (2010), 'Prediction of rbnbs and ibnr claims using claim amounts and claim counts', *ASTIN Bulletin: The Journal of the IAA* **40**(2), 871–887.
- Wahl, F., Lindholm, M. & Verrall, R. (2019), 'The collective reserving model', *Insurance: Mathematics and Economics* **87**, 34–50.
- Wüthrich, M. V. (2018), 'Neural networks applied to chain–ladder reserving', *European Actuarial Journal* **8**(2), 407–436.