Machine Learning, Regression Models, and Prediction of Claims Reserves

Mathias Lindholm
Stockholm university

CLRS, September, 2020

The majority of the presented material is based on recent and ongoing joint work with

- Richard Verrall
- Felix Wahl
- Henning Zakrisson

in particular the manuscript

"Machine Learning, Regression Models, and Prediction of Claims Reserves"

which is accepted for publication in CAS E-Forum, see Lindholm et al. (2020)

Aim for today

Introduce reserving models that can produce separate RBNS and IBNR reserves

Aim for today

Introduce reserving models that can produce separate RBNS and IBNR reserves...

...and discuss how these can be combined with machine learning techniques

Aim for today

Introduce reserving models that can produce separate RBNS and IBNR reserves...

...and discuss how these can be combined with machine learning techniques

Focus will be on general concepts and intuition rather than rigour

Aim for today

Introduce reserving models that can produce separate RBNS and IBNR reserves...

...and discuss how these can be combined with machine learning techniques

Focus will be on general concepts and intuition rather than rigour

(i.e. prepare for some abuse of notation, for more details see Lindholm et al. (2020))

Outline of the talk:

- Regression models: Double Chain-Ladder (DCL) type models
- Parametrisations and machine learning estimation
- Numerical illustrations: Gradient Boosting Machines and Neural Networks
- Closing remarks

Regression models – initial considerations

- Aggregated payment data used by e.g. the std chain-ladder technique is not sufficient in order to produce separate IBNR / RBNS reserves

Regression models – initial considerations

- Aggregated payment data used by e.g. the std chain-ladder technique is not sufficient in order to produce separate IBNR / RBNS reserves
- Highly detailed (continuous time) micro level data is often cumbersome to work with ("bespoke" models)

Regression models – initial considerations

- Aggregated payment data used by e.g. the std chain-ladder technique is not sufficient in order to produce separate IBNR / RBNS reserves
- Highly detailed (continuous time) micro level data is often cumbersome to work with ("bespoke" models)
- Compromise: Use discrete time aggregated incremental payment data and claim count data

Regression models – initial considerations

- Aggregated payment data used by e.g. the std chain-ladder technique is not sufficient in order to produce separate IBNR / RBNS reserves
- Highly detailed (continuous time) micro level data is often cumbersome to work with ("bespoke" models)
- Compromise: Use discrete time aggregated incremental payment data and claim count data

  $\Rightarrow$ "Double Chain-Ladder" (DCL) type models

Regression models – initial considerations

- Aggregated payment data used by e.g. the std chain-ladder technique is not sufficient in order to produce separate IBNR / RBNS reserves
- Highly detailed (continuous time) micro level data is often cumbersome to work with ("bespoke" models)
- Compromise: Use discrete time aggregated incremental payment data and claim count data

  $\Rightarrow$ "Double Chain-Ladder" (DCL) type models
- DCL type models are similar to standard chain-ladder technique models, that
  - can produce separate RBNS / IBNR reserves
  - have parameters with constructive interpretations
  - are easy to bootstrap

Regression models – DCL type models

Some notation

Regression models – DCL type models

Some notation: Let

- X_{ij} denote the incremental claims payments from accident year $i = 1, \ldots, m$ paid during development year $j = 0, \ldots, m - 1 + d$, where d corresponds to the maximal reporting delay since claim reporting

Regression models – DCL type models

Some notation: Let

- X_{ij} denote the incremental claims payments from accident year $i = 1, \ldots, m$ paid during development year $j = 0, \ldots, m - 1 + d$, where d corresponds to the maximal reporting delay since claim reporting
- N_{ij} denote the incremental number of claims from accident year $i = 1, \ldots, m$ reported during development year $j = 0, \ldots, m - 1$

Regression models – DCL type models

Some notation: Let

- X_{ij} denote the incremental claims payments from accident year $i = 1, \ldots, m$ paid during development year $j = 0, \ldots, m - 1 + d$, where d corresponds to the maximal reporting delay since claim reporting
- N_{ij} denote the incremental number of claims from accident year $i = 1, \ldots, m$ reported during development year $j = 0, \ldots, m - 1$
- $\mathcal{N}$ denote all information about the N_{ij}s corresponding to that contained in a standard claims triangle

Regression models – DCL type models

Some notation: Let

- X_{ij} denote the incremental claims payments from accident year $i = 1, \ldots, m$ paid during development year $j = 0, \ldots, m - 1 + d$, where d corresponds to the maximal reporting delay since claim reporting
- N_{ij} denote the incremental number of claims from accident year $i = 1, \ldots, m$ reported during development year $j = 0, \ldots, m - 1$
- $\mathcal{N}$ denote all information about the N_{ij}s corresponding to that contained in a standard claims triangle

We will not present the standard DCL model from Miranda et al. (2012), but rather the so-called Collective Reserving Model (CRM) from Wahl et al. (2019) which is an over-dispersed Poisson model

Regression models – DCL type models

Model 1

The CRM based on $X_{i,j}$ and $N_{i,j}$ data from Wahl et al. (2019) can be written on the form

$$X_{i,j} \mid \mathcal{N} \sim \mathrm{ODP}\left(\sum_{k=0}^{j\wedge d} \psi_{i,j-k,k} N_{i,j-k},\ \varphi\right),$$

and

$$N_{i,j} \sim \mathrm{ODP}\left(\nu_{i,j},\ \phi\right),$$

that is,

$$\mathrm{E}[X_{i,j} \mid \mathcal{N}] = \sum_{k=0}^{j\wedge d} \psi_{i,j-k,k} N_{i,j-k} = \mathrm{Var}(X_{i,j} \mid \mathcal{N})/\varphi,$$

and

$$\mathrm{E}[N_{i,j}] = \nu_{i,j} = \mathrm{Var}(N_{i,j})/\phi.$$

All $X_{i,j}$ are assumed to be conditionally independent, given $\mathcal{N}$, and all $N_{i,j}$ are assumed to be independent.

Regression models – DCL type models

Regression models – DCL type models

(i) φ and ϕ are over-dispersion parameters – setting these to 1 is equivalent to standard Poisson

Regression models – DCL type models

(i) φ and ϕ are over-dispersion parameters – setting these to 1 is equivalent to standard Poisson

(ii) The N_{ij}s act as exposures in the X_{ij} part of the model, and the N_{ij}s are modelled following a standard ODP cross-classified Chain-Ladder model

Regression models – DCL type models

(i) φ and ϕ are over-dispersion parameters – setting these to 1 is equivalent to standard Poisson

(ii) The N_{ij}s act as exposures in the X_{ij} part of the model, and the N_{ij}s are modelled following a standard ODP cross-classified Chain-Ladder model

(iii) The ψ_{ijk}s describe the average amount paid for claims from accident year i that where reported in development period j, and paid k periods after reporting

Regression models – DCL type models

(i) φ and ϕ are over-dispersion parameters – setting these to 1 is equivalent to standard Poisson

(ii) The N_{ij}s act as exposures in the X_{ij} part of the model, and the N_{ij}s are modelled following a standard ODP cross-classified Chain-Ladder model

(iii) The ψ_{ijk}s describe the average amount paid for claims from accident year i that where reported in development period j, and paid k periods after reporting

(iv) The RBNS reserves are obtained by conditioning on knowledge of N_{ij}s, and the IBNR reserves are obtained by also predicting future N_{ij}s

Regression models – DCL type models

(i) φ and ϕ are over-dispersion parameters – setting these to 1 is equivalent to standard Poisson

(ii) The N_{ij}s act as exposures in the X_{ij} part of the model, and the N_{ij}s are modelled following a standard ODP cross-classified Chain-Ladder model

(iii) The ψ_{ijk}s describe the average amount paid for claims from accident year i that where reported in development period j, and paid k periods after reporting

(iv) The RBNS reserves are obtained by conditioning on knowledge of N_{ij}s, and the IBNR reserves are obtained by also predicting future N_{ij}s, i.e. (for combinations of (i, j) corresponding to the future, with allowed values of k)

RBNS contribution: $\mathrm{E}[X_{i,j} \mid \mathcal{N}] = \sum_k \psi_{i,j-k,k} N_{i,j-k}$

IBNR contribution: $\mathrm{E}[X_{i,j}] = \sum_k \psi_{i,j-k,k} \mathrm{E}[N_{i,j-k}]$

Regression models – DCL type models

Note that (iii) discusses (i,j,k)-information, which is something needed in the derivation of the CRM (and DCL):

- Let X_{ijk} denote the total amount of payments from accident year i, that come from claims that are reported with j years delay, and that are paid k years after reporting

Regression models – DCL type models

Note that (iii) discusses (i,j,k)-information, which is something needed in the derivation of the CRM (and DCL):

- Let X_{ijk} denote the total amount of payments from accident year i, that come from claims that are reported with j years delay, and that are paid k years after reporting...
- ...but when using $X_{i,j,k}$ data the $N_{i,j}$s are not needed to produce separate RBNS and IBNR reserves, since the outstanding claim payments

$$R_i^{\mathcal{R}} := \sum_{j=0}^{m-i} \sum_{k>m-(i+j)} X_{i,j,k},$$

and

$$R_i^{\mathcal{I}} := \sum_{j=m-i+1}^{m-1} \sum_{k} X_{i,j,k}.$$

Regression models – DCL type models

This suggests two additional models:

Model 2

The CRM based on $X_{i,j,k}$ and $N_{i,j}$ data from Wahl et al. (2019) can be written on the form

$$X_{i,j,k} \mid \mathcal{N} \sim \mathrm{ODP}\left(\psi_{i,j,k} N_{i,j},\ \varphi\right),$$

and

$$N_{i,j} \sim \mathrm{ODP}\left(\nu_{i,j},\ \phi\right),$$

and in particular,

$$\mathrm{E}[X_{i,j,k} \mid \mathcal{N}] = \psi_{i,j,k} N_{i,j},$$

All $X_{i,j,k}$ are assumed to be conditionally independent, given $\mathcal{N}$, and all $N_{i,j}$ are assumed to be independent.

Model 3

Let $X_{i,j,k}$ be defined as an over-dispersed model according to

$$X_{i,j,k} \sim \mathrm{ODP}\left(\psi_{i,j,k},\ \varphi\right),$$

with

$$\mathrm{E}[X_{i,j,k}] = \psi_{i,j,k} = \mathrm{Var}(X_{i,j,k})/\varphi,$$

where all $X_{i,j,k}$ are assumed to be independent.

Parametrisations and machine learning estimation

- Model 1 – 3 are very general regression models and consequently hard to estimate unless restrictions are imposed on the ψ_{ijk}s

Parametrisations and machine learning estimation

- Model 1 – 3 are very general regression models and consequently hard to estimate unless restrictions are imposed on the ψ_{ijk}s
- E.g. $\psi_{ijk} := \gamma_k, \psi_{ijk} := \alpha_{i+j}\gamma_k$ or $\psi_{ijk} := \alpha_i\beta_j\gamma_k$

Parametrisations and machine learning estimation

- Model 1 – 3 are very general regression models and consequently hard to estimate unless restrictions are imposed on the ψ_{ijk}s
- E.g. $\psi_{ijk} := \gamma_k, \psi_{ijk} := \alpha_{i+j}\gamma_k$ or $\psi_{ijk} := \alpha_i\beta_j\gamma_k$...
- ...or $\psi_{ijk} := f(i, j, k; \boldsymbol{\theta})$, where $f(\cdot)$ is some suitably chosen function

Parametrisations and machine learning estimation

- Model 1 – 3 are very general regression models and consequently hard to estimate unless restrictions are imposed on the ψ_{ijk}s
- E.g. $\psi_{ijk} := \gamma_k, \psi_{ijk} := \alpha_{i+j}\gamma_k$ or $\psi_{ijk} := \alpha_i\beta_j\gamma_k$...
- ...or $\psi_{ijk} := f(i, j, k; \boldsymbol{\theta})$, where $f(\cdot)$ is some suitably chosen function
- ... "Machine Learning"

Parametrisations and machine learning estimation

- Model 1 – 3 are very general regression models and consequently hard to estimate unless restrictions are imposed on the ψ_{ijk}s
- E.g. $\psi_{ijk} := \gamma_k, \psi_{ijk} := \alpha_{i+j}\gamma_k$ or $\psi_{ijk} := \alpha_i\beta_j\gamma_k$...
- ...or $\psi_{ijk} := f(i, j, k; \boldsymbol{\theta})$, where $f(\cdot)$ is some suitably chosen function
- ... "Machine Learning": Use a very flexible class of functions $f(\cdot)$ often defined (implicitly) using a very large number of parameters optimised w.r.t. a suitable loss function

Parametrisations and machine learning estimation

- Model 1 – 3 are very general regression models and consequently hard to estimate unless restrictions are imposed on the ψ_{ijk}s
- E.g. $\psi_{ijk} := \gamma_k, \psi_{ijk} := \alpha_{i+j}\gamma_k$ or $\psi_{ijk} := \alpha_i\beta_j\gamma_k$...
- ...or $\psi_{ijk} := f(i, j, k; \boldsymbol{\theta})$, where $f(\cdot)$ is some suitably chosen function
- ... "Machine Learning": Use a very flexible class of functions $f(\cdot)$ often defined (implicitly) using a very large number of parameters optimised w.r.t. a suitable loss function

  Here: $f(\cdot)$ will be given by Gradient Boosting Machines (GBM) and Neural Networks (NN) optimised w.r.t. a Poisson loss (relating to the likelihood for Model 1 – 3)

Parametrisations and machine learning estimation

- Model 1 – 3 are very general regression models and consequently hard to estimate unless restrictions are imposed on the ψ_{ijk}s
- E.g. $\psi_{ijk} := \gamma_k, \psi_{ijk} := \alpha_{i+j}\gamma_k$ or $\psi_{ijk} := \alpha_i\beta_j\gamma_k$...
- ...or $\psi_{ijk} := f(i, j, k; \boldsymbol{\theta})$, where $f(\cdot)$ is some suitably chosen function
- ... "Machine Learning": Use a very flexible class of functions $f(\cdot)$ often defined (implicitly) using a very large number of parameters optimised w.r.t. a suitable loss function

  Here: $f(\cdot)$ will be given by Gradient Boosting Machines (GBM) and Neural Networks (NN) optimised w.r.t. a Poisson loss (relating to the likelihood for Model 1 – 3)
- **Problem:** overfitting

Parametrisations and machine learning estimation – Calibration and overfitting

- GBMs and NNs are estimated using iterative numerical procedures

Parametrisations and machine learning estimation – Calibration and overfitting

- GBMs and NNs are estimated using iterative numerical procedures
- **Ideally:** for each iteration in the optimisation
  1. estimate all parameters based on *complete* in-sample (training) data and evaluate the estimates on *complete* out-of-sample (validation) data

Parametrisations and machine learning estimation – Calibration and overfitting

- GBMs and NNs are estimated using iterative numerical procedures
- **Ideally:** for each iteration in the optimisation
  1. estimate all parameters based on *complete* in-sample (training) data and evaluate the estimates on *complete* out-of-sample (validation) data
  2. choose the iteration that performs best w.r.t. out-of-sample performance

Parametrisations and machine learning estimation – Calibration and overfitting

- GBMs and NNs are estimated using iterative numerical procedures
- **Ideally:** for each iteration in the optimisation
  1. estimate all parameters based on *complete* in-sample (training) data and evaluate the estimates on *complete* out-of-sample (validation) data
  2. choose the iteration that performs best w.r.t. out-of-sample performance
- This procedure is known as "early stopping"

Parametrisations and machine learning estimation – Calibration and overfitting

- GBMs and NNs are estimated using iterative numerical procedures
- **Ideally:** for each iteration in the optimisation
  1. estimate all parameters based on *complete* in-sample (training) data and evaluate the estimates on *complete* out-of-sample (validation) data
  2. choose the iteration that performs best w.r.t. out-of-sample performance
- This procedure is known as "early stopping"
- **Problem:** In a claims reserving context we don't have access to fully developed triangles that are representative for the future

Parametrisations and machine learning estimation – Calibration and overfitting

Following Gabrielli et al. (2019) we instead start from the *claims database* and construct

(i) one set of *historical* triangles to be used as “in-sample” data for the purpose of estimation

(ii) one set of *historical* triangles to be used as “out-of-sample” data for the purpose of evaluation

Parametrisations and machine learning estimation – Calibration and overfitting

Following Gabrielli et al. (2019) we instead start from the *claims database* and construct

(i) one set of *historical* triangles to be used as "in-sample" data for the purpose of estimation

(ii) one set of *historical* triangles to be used as "out-of-sample" data for the purpose of evaluation

Note that

- Using these triangles (of X_{ij}, X_{ijk} and N_{ij} data depending on model) the procedure from the previous slide can be used

Parametrisations and machine learning estimation – Calibration and overfitting

Following Gabrielli et al. (2019) we instead start from the *claims database* and construct

(i) one set of *historical* triangles to be used as "in-sample" data for the purpose of estimation

(ii) one set of *historical* triangles to be used as "out-of-sample" data for the purpose of evaluation

Note that

- Using these triangles (of X_{ij}, X_{ijk} and N_{ij} data depending on model) the procedure from the previous slide can be used
- This procedure is not depending on any specific choice of machine learning technique

Parametrisations and machine learning estimation – Calibration and overfitting

Following Gabrielli et al. (2019) we instead start from the *claims database* and construct

(i) one set of *historical* triangles to be used as "in-sample" data for the purpose of estimation

(ii) one set of *historical* triangles to be used as "out-of-sample" data for the purpose of evaluation

Note that

- Using these triangles (of X_{ij}, X_{ijk} and N_{ij} data depending on model) the procedure from the previous slide can be used
- This procedure is not depending on any specific choice of machine learning technique

Next: numerical illustration using GBMs and NNs

Numerical illustrations

The models to be considered are (primarily)

- The (homogeneous) CRM: $\psi_{ijk} = \gamma_k$
  (special case of Model 1)
- The Generalized CRM (GCRM): $\psi_{ijk} = \alpha_i \beta_j \gamma_k$
  (special case of Model 2)
- Model 2 and 3 estimated using GBMs and (feed-forward) NNs

Numerical illustrations

The models to be considered are (primarily)

- The (homogeneous) CRM: $\psi_{ijk} = \gamma_k$
  (special case of Model 1)
- The Generalized CRM (GCRM): $\psi_{ijk} = \alpha_i \beta_j \gamma_k$
  (special case of Model 2)
- Model 2 and 3 estimated using GBMs and (feed-forward) NNs

Moreover,

- We consider six different portfolios of simulated data generated by the individual claims history simulation machine of Gabrielli and Wüthrich (2018), which is a NN model calibrated to Swiss insurance data

Numerical illustrations

The models to be considered are (primarily)

- The (homogeneous) CRM: $\psi_{ijk} = \gamma_k$ (special case of Model 1)
- The Generalized CRM (GCRM): $\psi_{ijk} = \alpha_i \beta_j \gamma_k$ (special case of Model 2)
- Model 2 and 3 estimated using GBMs and (feed-forward) NNs

Moreover,

- We consider six different portfolios of simulated data generated by the individual claims history simulation machine of Gabrielli and Wüthrich (2018), which is a NN model calibrated to Swiss insurance data
- For reference we will use the standard chain-ladder technique (CL) and the ODP CL NN model from Gabrielli et al. (2019) (GRWNN)

Numerical illustrations

The models to be considered are (primarily)

- The (homogeneous) CRM: $\psi_{ijk} = \gamma_k$ (special case of Model 1)
- The Generalized CRM (GCRM): $\psi_{ijk} = \alpha_i \beta_j \gamma_k$ (special case of Model 2)
- Model 2 and 3 estimated using GBMs and (feed-forward) NNs

Moreover,

- We consider six different portfolios of simulated data generated by the individual claims history simulation machine of Gabrielli and Wüthrich (2018), which is a NN model calibrated to Swiss insurance data
- For reference we will use the standard chain-ladder technique (CL) and the ODP CL NN model from Gabrielli et al. (2019) (GRWNN)
- All implementations are done using standard `R`-packages (`keras` and `gbm`)

Table: True reserve compared to CL, the CRM and GRWNN reserves. Relative bias (error) of the predictions in the parentheses. The data is the same as that used in Gabrielli et al. (2019).

	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	35,460	15,692	67,574	70,166	29,409
	(-2.82)	(-4.26)	(-7.02)	(-5.66)	(-3.28)	(-5.49)
CRM reserves	32,485	29,901	13,040	55,782	59,390	24,403
	(-18.15)	(-19.27)	(-22.74)	(-22.12)	(-18.14)	(-21.58)
GCRM reserves	38,293	35,117	15,448	66,961	69,397	29,104
	(-3.52)	(-5.18)	(-8.47)	(-6.52)	(-4.34)	(-6.47)
GRWNN	39,233	35,899	15,815	70,219	70,936	30,671
	(-1.15)	(-3.07)	(-6.30)	(-1.97)	(-2.22)	(-1.43)

The relative bias is calculated based on that we know the true simulated payment outcome

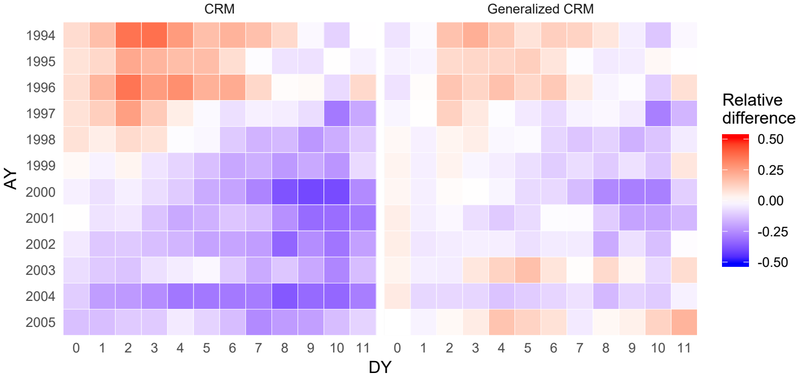

Figure: Heatmap for the relative biases (errors) of the total payment prediction within a specific accident year and development year combination for LoB 1 using the CRM and GCRM.

Numerical illustrations

- So there seems to be room for improvement for the RBNS / IBNR split models!

Numerical illustrations

- So there seems to be room for improvement for the RBNS / IBNR split models!
- Focus on GBMs

Numerical illustrations

- So there seems to be room for improvement for the RBNS / IBNR split models!
- Focus on GBMs – parameters to be tuned:
  - Shrinkage factor (or learning rate)
  - Bagging fraction
  - ...

Numerical illustrations

- So there seems to be room for improvement for the RBNS / IBNR split models!
- Focus on GBMs – parameters to be tuned:
  - Shrinkage factor (or learning rate)
  - Bagging fraction
  - ...
- The important part here is not what these parameters are, but rather to illustrate how this may be done

Numerical illustrations

- So there seems to be room for improvement for the RBNS / IBNR split models!
- Focus on GBMs – parameters to be tuned:
  - Shrinkage factor (or learning rate)
  - Bagging fraction
  - ...
- The important part here is not what these parameters are, but rather to illustrate how this may be done
- That is, to decide on the number of trees to be used (= no. of iterations in the numerical optimisation) for a given set of tuning parameters

Numerical illustrations

- So there seems to be room for improvement for the RBNS / IBNR split models!
- Focus on GBMs – parameters to be tuned:
  - Shrinkage factor (or learning rate)
  - Bagging fraction
  - ...
- The important part here is not what these parameters are, but rather to illustrate how this may be done
- That is, to decide on the number of trees to be used (= no. of iterations in the numerical optimisation) for a given set of tuning parameters
- Loss function used is the unscaled Poisson deviance

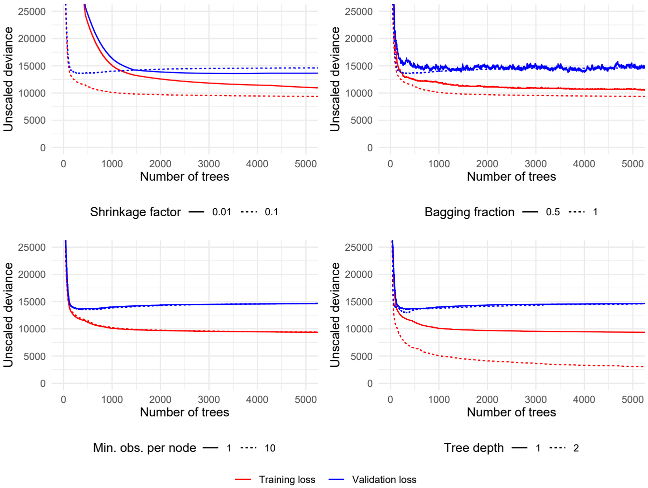

Figure: Training and validation loss for our tuning of the GBM for the payment part of Model 2 when varying the shrinkage factor (upper left),the bagging fraction (upper right), the minimum number of observations per node (lower left), and the tree depth (lower right).

Numerical illustrations

- In the remainder we use shrinkage 0.1, bagging fraction 1, min. observation per node 1, and tree depth 1 or 2, depending on LoB

Numerical illustrations

- In the remainder we use shrinkage 0.1, bagging fraction 1, min. observation per node 1, and tree depth 1 or 2, depending on LoB
- Number of trees: 200 and 900 trees, depending on LoB

Numerical illustrations

- In the remainder we use shrinkage 0.1, bagging fraction 1, min. observation per node 1, and tree depth 1 or 2, depending on LoB
- Number of trees: 200 and 900 trees, depending on LoB
- We use an analogous procedure for the NN models, see Gabrielli et al. (2019); Lindholm et al. (2020)

Numerical illustrations

- In the remainder we use shrinkage 0.1, bagging fraction 1, min. observation per node 1, and tree depth 1 or 2, depending on LoB
- Number of trees: 200 and 900 trees, depending on LoB
- We use an analogous procedure for the NN models, see Gabrielli et al. (2019); Lindholm et al. (2020)
- The best performing model turns out to be Model 2 which is illustrated next using 100 simulated datasets, see Lindholm et al. (2020) for more details

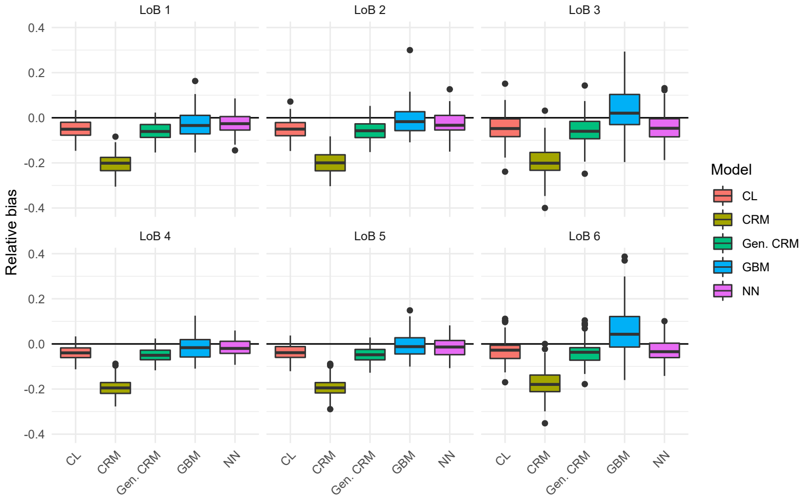

Figure: Boxplots of relative biases for total reserves from 100 simulations. Here GBM and NN refers to Model 2.

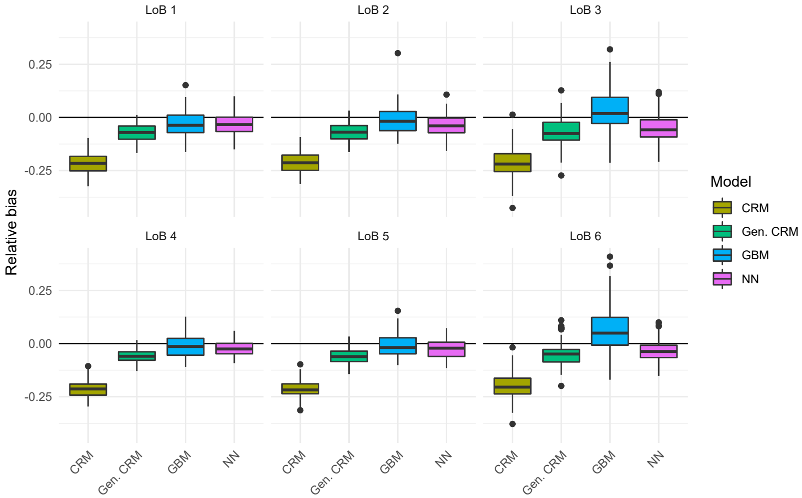

Figure: Boxplots of relative biases for RBNS reserves from 100 simulations. Here GBM and NN refers to Model 2.

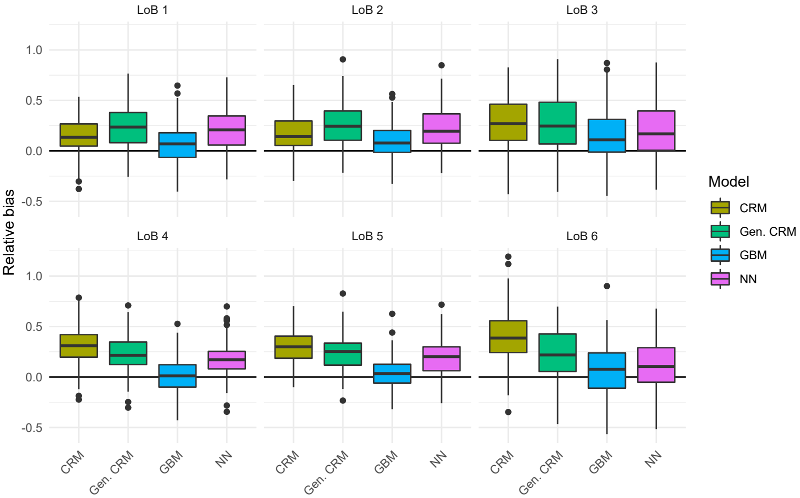

Figure: Boxplots of relative biases for IBNR reserves from 100 simulations. Here GBM and NN refers to Model 2.

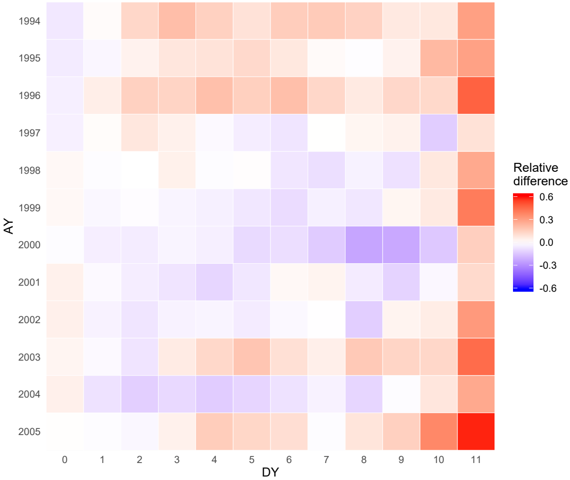

Figure: Heatmap for the relative biases (errors) of the total payment prediction within a specific accident year and development year combination for LoB 1 using the GBM – Model 2.

Closing remarks

Closing remarks

- We have introduced regression based reserving models that can be used to produce separate RBNS and IBNR reserves

Closing remarks

- We have introduced regression based reserving models that can be used to produce separate RBNS and IBNR reserves
- We have described how these models can be estimated using machine learning techniques with illustrations on complex simulated data

Closing remarks

- We have introduced regression based reserving models that can be used to produce separate RBNS and IBNR reserves
- We have described how these models can be estimated using machine learning techniques with illustrations on complex simulated data
- The numerical illustrations show how the use of machine learning techniques may improve the predictive performance

Closing remarks

- We have introduced regression based reserving models that can be used to produce separate RBNS and IBNR reserves
- We have described how these models can be estimated using machine learning techniques with illustrations on complex simulated data
- The numerical illustrations show how the use of machine learning techniques may improve the predictive performance
- All implementations are done using standard publicly available software packages (in R)

Closing remarks

- We have introduced regression based reserving models that can be used to produce separate RBNS and IBNR reserves
- We have described how these models can be estimated using machine learning techniques with illustrations on complex simulated data
- The numerical illustrations show how the use of machine learning techniques may improve the predictive performance
- All implementations are done using standard publicly available software packages (in `R`)

For more details about the regression modelling aspects, see e.g. Wahl et al. (2019); Lindholm and Verrall (2020), and for a more detailed account of the presentation (including MSEP), see Lindholm et al. (2020)

References

Gabrielli, A., Richman, R., and Wüthrich, M. V. (2019). Neural network embedding of the over-dispersed poisson reserving model. *Scandinavian Actuarial Journal*, pages 1–29.

Gabrielli, A. and Wüthrich, M. V. (2018). An individual claims history simulation machine. *Risks*, 6(2):29.

Lindholm, M. and Verrall, R. (2020). Regression based reserving models and partial information. *Accepted for publication in Insurance: Mathematics and Economics*.

Lindholm, M., Verrall, R., Wahl, F., and Zakrisson, H. (2020). Machine learning, regression models, and prediction of claims reserves. *Accepted for publication in CAS E-Forum*.

Miranda, M. D. M., Nielsen, J. P., and Verrall, R. (2012). Double chain ladder. *ASTIN Bulletin: The Journal of the IAA*, 42(1):59–76.

Wahl, F., Lindholm, M., and Verrall, R. (2019). The collective reserving model. *Insurance: Mathematics and Economics*, 87:34–50.

Appendix

Table: True reserve compared to benchmark models and the GBMs and NNs. Relative biases of the reserve predictions in the parentheses. The data is the same as that used in Gabrielli et al. (2019).

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