Building a Reserving Robot

Gráinne McGuire

Building a reserving robot

- Contents
  - – Introduction
  - – Reserving robots – who needs them?
  - – What type of robot?
  - – Adaptive filters
  - – Programming the robot
  - – The robot in action
  - – Discussion

Introduction

- What is a reserving robot?
  - An automatic process for carrying out claims reserving
  - Software, not hardware

Reserving robots – who needs them?

- Why build a robot?
  - We spend many years learning to do outstanding claims reserving.
  - Shouldn't we actually use all that experience?
  - Yes, but…
    - Why spend time slogging over routine reserving jobs?
    - Why not use that time for difficult reserving problems?

    (Unless, of course, you enjoy fitting endless chain ladders, PPCIs, PCEs etc.)

Reserving robots – who needs them?

- Imagine you do the reserving for a large general insurance company
  - You have a large number of lines of business (LOBs)
  - Valuations tend to happen on a revolving door basis – one finishes and the next one begins
  - You (or your minions) spend much of your time fitting reserving models to each LOB
  - A robot can change this

Reserving robots – who needs them?

- Imagine you have a robot
  - Most LOBs will show little change from one valuation period to the next
  - Some LOBs will require minor adjustments to modelling assumptions, a few might require more substantial changes
  - So why not let your robot deal with the routine LOBs, leaving you free to focus on those LOBs that need your experience?

What type of robot?

- We need a robot that can
  - Apply a model to data
  - Adapt this model if warranted by recent experience
  - Evaluate the model's performance
  - Project results, including central estimates and risk margins
- What would do all that?

Adaptive filters

- Dynamical statistical models
  - adapt over time to changing experience
  - may be tested for goodness of fit
  - Distributional information is available
- A suitable dynamic model might be an adaptive filter

Adaptive filters

- Kalman filter
  - Been around for a while (1960, actuarial literature since 1983)
  - A form of time series estimation in which parameter estimates are constructed so as to track evolving parameters
  - The model for each epoch is that based on data up to the start of the epoch, modified by the experience of the new data.

Adaptive filters

- Kalman filter

Forecast new epoch's parameters and observations **without** new information

$$\beta_{j+1|j} = G_{j+1} \beta_{j|j}$$

$$\Gamma_{j+1|j} = G_{j+1} \Gamma_{j|j} G^{T}_{j+1} + W_{j+1}$$

$$Y_{j+1|j} = X_{j+1} \beta_{j+1|j}$$

Calculate gain matrix (credibility of new observation)

$$L_{j+1|j} = X_{j+1} \Gamma_{j+1|j} X^{T}_{j+1} + V_{j+1}$$

$$K_{j+1} = \Gamma_{j+1|j} X^{T}_{j+1} [L_{j+1|j}]^{-1}$$

Update parameter estimates to incorporate new observation

$$\beta_{j+1|j+1} = \beta_{j+1|j} + K_{j+1} (Y_{j+1} - Y_{j+1|j})$$

$$\Gamma_{j+1|j+1} = (1 - K_{j+1} X_{j+1}) \Gamma_{j+1|j}$$

Adaptive filters

- The Kalman filter
  - Is a statistical model
  - Is fast since it is analytical
  - Can adapt to changing experience
  - Can be used in a reserving robot

Adaptive filters

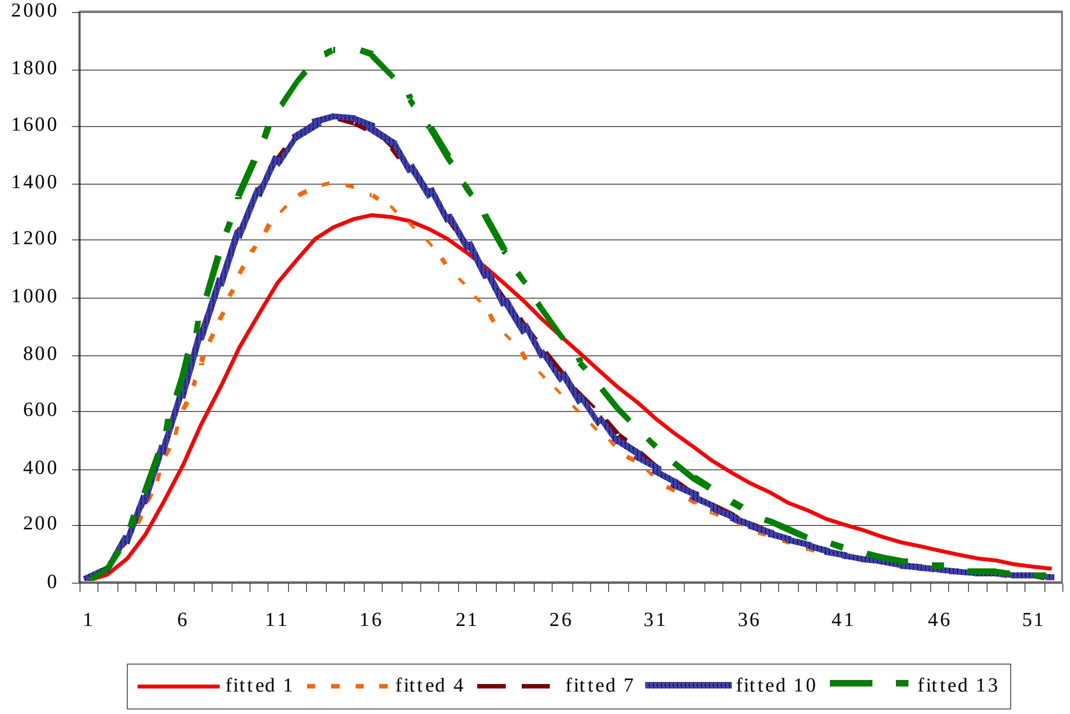
- But the Kalman filter
  - Requires normally distributed data
    - So typical assumption of log normal claim payments made
    - But this can be problematic – e.g. estimation of bias correction
    - Further some things are not naturally represented by a normal or log-normal distribution (e.g. finalisation numbers, claim counts)
  - Any alternatives?

Adaptive filters

- GLM filters
  - Introduced by Taylor (2008)
  - Generalisation of the Kalman filter to some members of the exponential dispersion family
  - Analytical filter – based on second order approximations to Bayesian revision – for
    - Gamma error + (log link or reciprocal link)
    - Poisson error + log link
    - Normal error + identity link (=Kalman filter)

Adaptive filters

- What do they do?



Adaptive filters

- How do they do it?
  - Bayesian process
  - Essentially it comes down to the relative sizes of the parameter and data variance
    - Low parameter variance: parameters not expected to change much from one period to next and vice versa
    - Low data variance (noise): the fitted curves expected to closely fit the data and vice versa
  - Filters compare the relative sizes of these two variance components and fitted curves move accordingly

Programming the robot

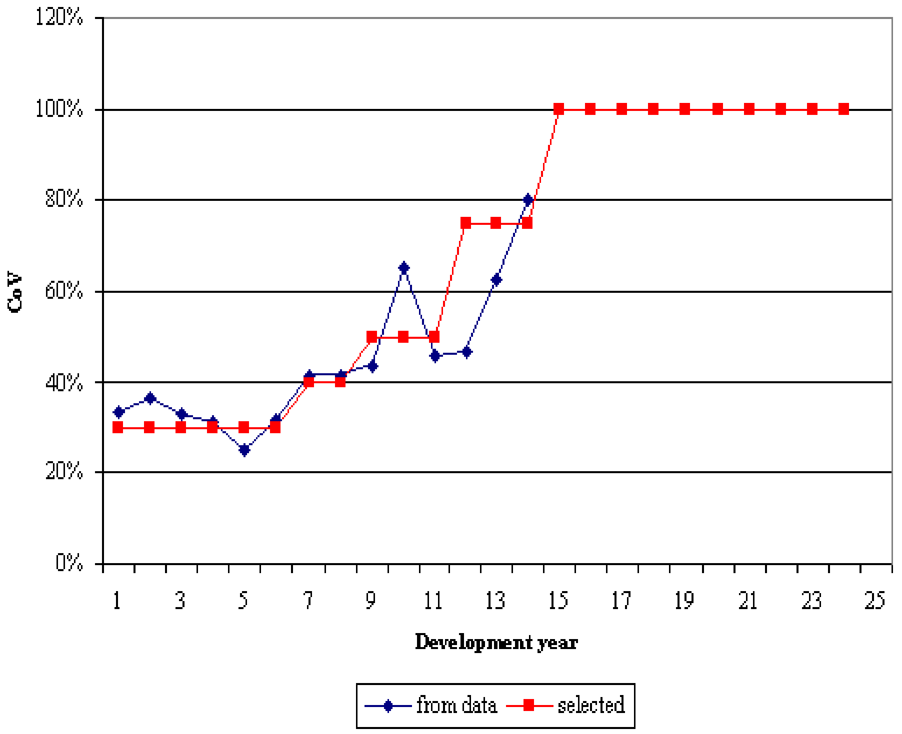
- Take some motor bodily injury data (from Taylor 2000) for years 1980 - 1995
- Let's set up a PPCI adaptive filter ($PPCI_{ij}$ = gross payments in accident year i and development year j / total claims in accident year i)
- What are the steps to programme the robot?

Programming the robot

- Step 1: select error + link
  - Claim size distribution of strictly positive claim sizes
  - Use Gamma error – no need to transform claim sizes
  - Log link ensures positive claim sizes

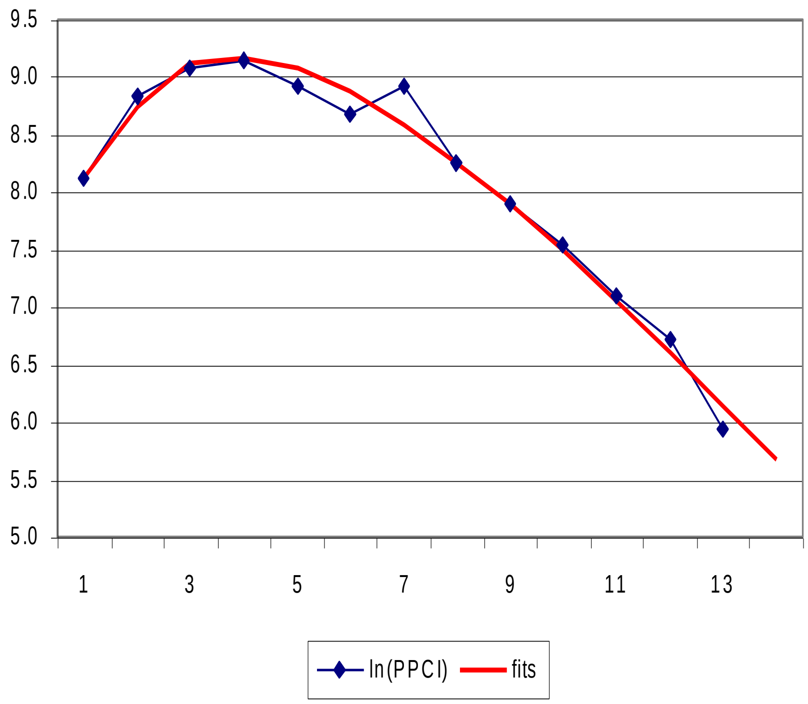
Programming the robot

- Step 2: process error assumptions
  - Error or variance of data
  - Set assumptions using coefficient of variation – assume it varies by development year only



Programming the robot

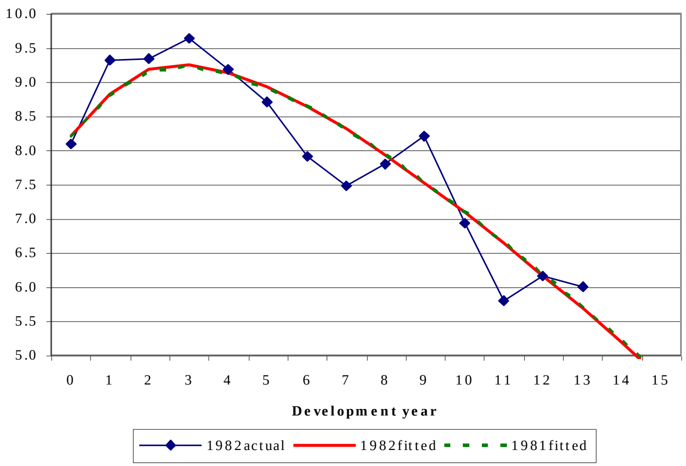
- Step 3: basis functions and parameter estimates
  - Hoerl curve
  - $\mu_{ij} = \exp\{\beta_0 + \beta_1(j-1) + \beta_2 \log(j) + \beta_3 I(j=1)$
  - Initial values for β_k needed.
    - Fit curve to average 83-85 experience (80-82 experience different to rest)

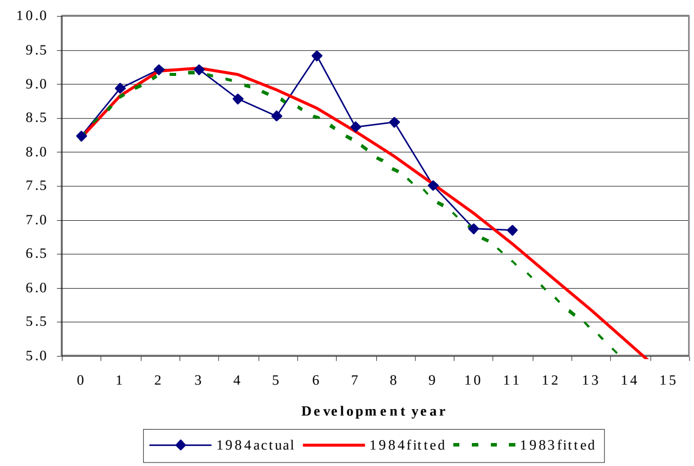


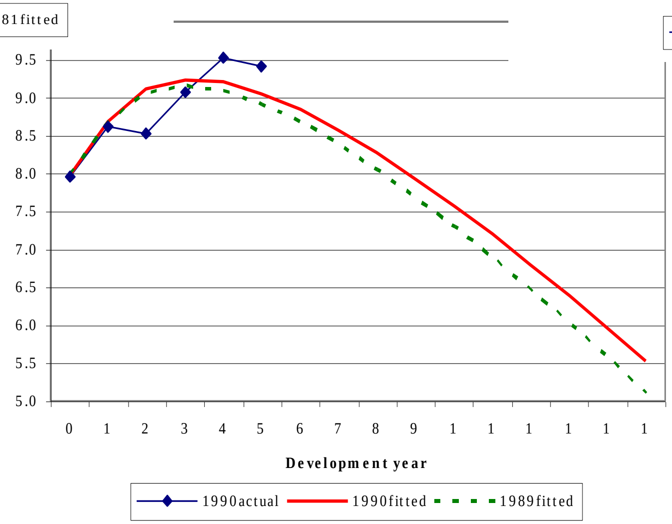
Programming the robot

- Step 4: parameter variance
  - Set by judgement at a level relative to data noise that ensures
    - The parameters don't move too much (overfit)
    - The parameters can move (ie model is adaptive not static)
  - Values in the range 10^{-3} to 10^{-5} suitable.
  - β_0 has variance of 0.001
    - the level of the PPCI curve has about a two thirds chance of not shifting by more than 3% from one accident period to the next.

Programming the robot



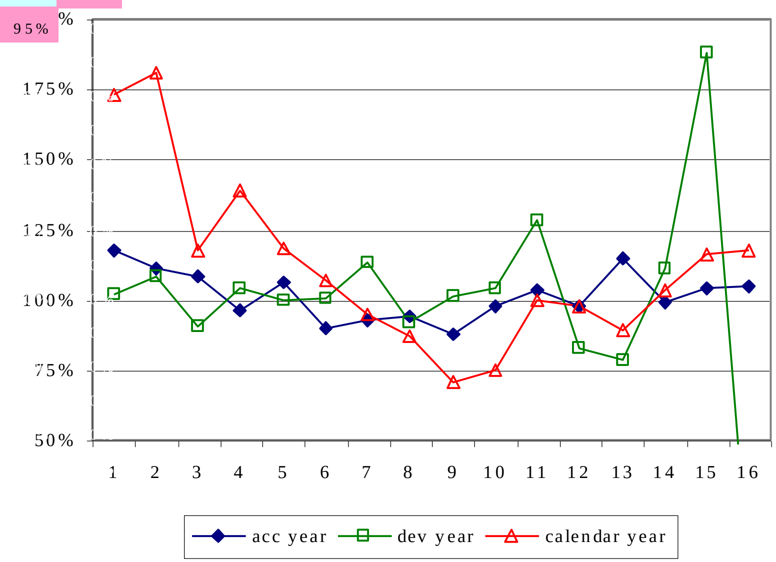




Programming the robot

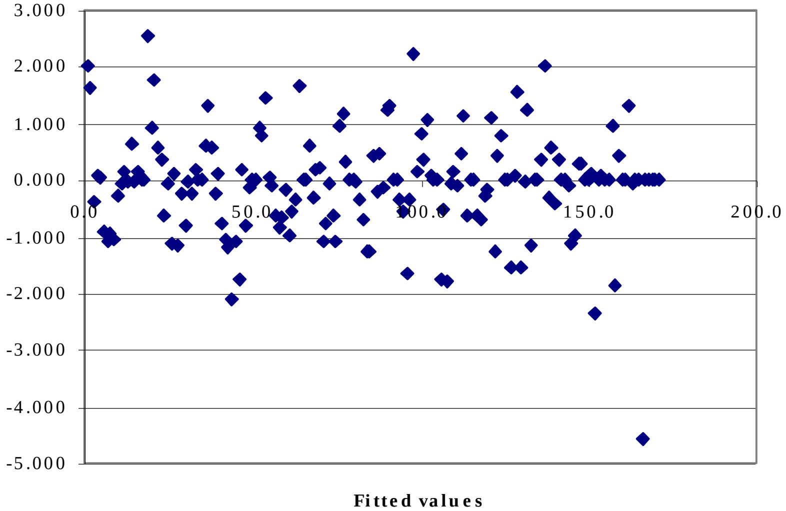
- Diagnostics

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1980	173%	196%	119%	130%	120%	99%	86%	45%	72%	41%	118%	92%	109%	189%	290%	0%
1981	158%	131%	145%	125%	91%	89%	79%	81%	69%	104%	236%	37%	9%	16%	95%	
1982	89%	162%	118%	150%	106%	80%	48%	44%	86%	199%	85%	43%	100%	137%		
1983	102%	118%	93%	102%	106%	67%	87%	98%	91%	41%	131%	122%	107%			
1984	101%	111%	103%	97%	70%	67%	216%	106%	166%	98%	81%	124%				
1985	75%	82%	79%	82%	79%	113%	106%	96%	50%	175%	119%					
1986	71%	98%	71%	77%	98%	94%	136%	120%	122%	53%						
1987	74%	70%	68%	81%	83%	115%	114%	152%	144%							
1988	71%	104%	49%	95%	71%	96%	149%	77%								
1989	92%	69%	71%	73%	131%	142%	103%									
1990	98%	93%	56%	84%	139%	144%										
1991	104%	77%	106%	90%	109%											
1992	99%	99%	78%	159%												
1993	120%	93%	96%													
1994	100%	106%														
1995	105%															



Programming the robot

- Other typical residual plots also useful



Programming the robot

- Step 5: risk margins
  - Stochastic model so possible to get distributional information
  - Use the bootstrap
    - Residuals from filter not independent so bootstrap process must be modified to allow for that
    - Stoffer and Wall (1991) give method for Kalman filter. Appropriate modification for GLM filter given in McGuire and Taylor (2007).

Programming the robot

- Results from bootstrap

Accident year	Liability estimate $'(000)	Standard Deviation $'(000)	Coefficient of variation %	75-percentile (% of mean) %
1980	135	69	51	128
1981	244	128	52	140
1982	388	253	65	124
1983	498	317	64	123
1984	1,166	842	72	116
1985	1,912	1,390	73	121
1986	2,947	1,640	56	140
1987	5,285	2,837	54	130
1988	6,858	3,743	55	116
1989	12,149	5,490	45	120
1990	20,205	8,388	42	118
1991	28,910	11,683	40	115
1992	44,442	14,203	32	118
1993	52,551	15,142	29	114
1994	61,467	16,905	28	114
1995	68,180	17,576	26	111
Total	307,337	91,171	30	113

Programming the robot

- Step 6: blending of model results
  - Common to apply several models
    - E.g. PPCI, PPCF, PCE
    - These results are then blended
  - Algorithm used to generate weights that
    - The smoothness of the ratio of blended liability to current case estimates
    - The smoothness of the progression of the weights

Programming the robot

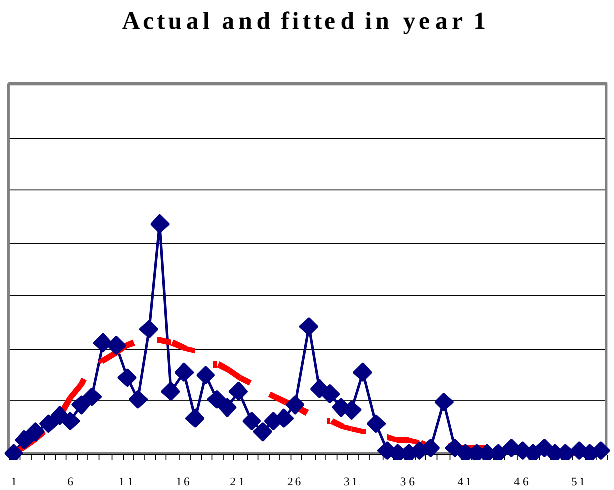
- Running the robot
  - – Big modelling effort first time round
  - – Thereafter “push of button” IF no major changes to data
    - Very important to carefully check diagnostics for any problems
  - – Model can adapt to changing experience
    - Major changes (eg due to legislation change) may require intervention

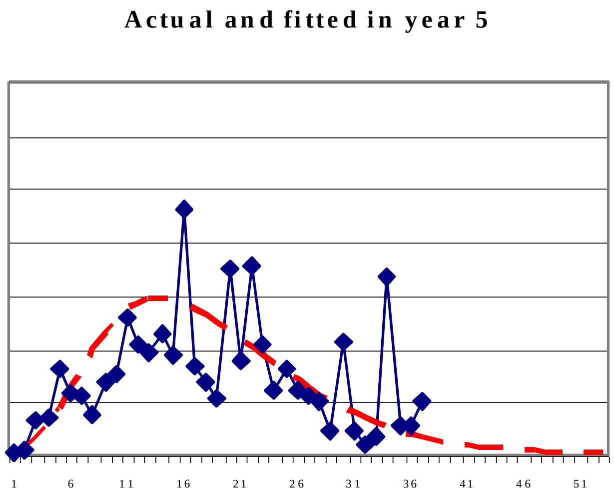
The robot in action

- Long tailed data
  - 14 accident periods, 50+ development periods
  - Split into two jurisdictions
  - PPCI, PPCF and PCE models built
  - Results bootstrapped and blended leading to
    - Liability estimate
    - Risk margin estimate
  - Results shown here for one jursidiction only

The robot in action

- PPCI model





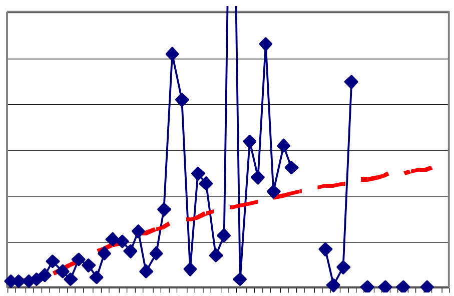
The robot in action

- PPCF model

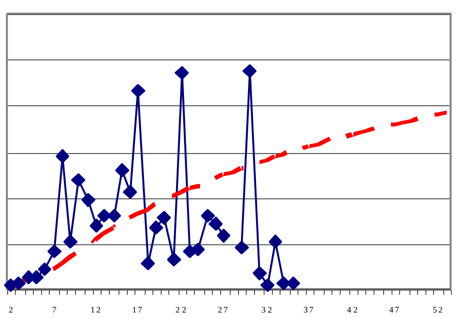
PPCF submodel

Finalisation rates submodel

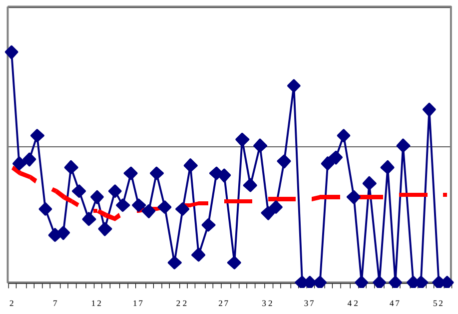
Actual and fitted in year 1



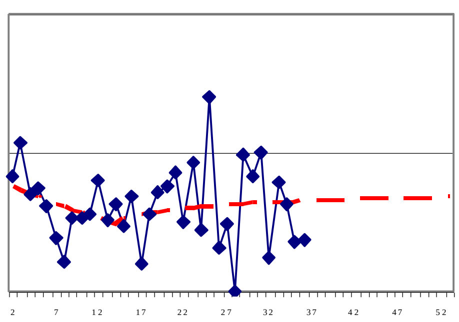
Actual and fitted in year 5



Actual and fitted in year 1



Actual and fitted in year 5

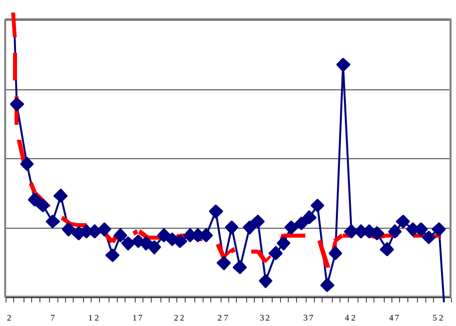


The robot in action

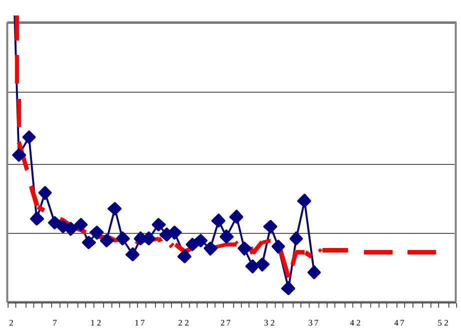
- PCE model

CEDF submodel

Actual and fitted in year 1

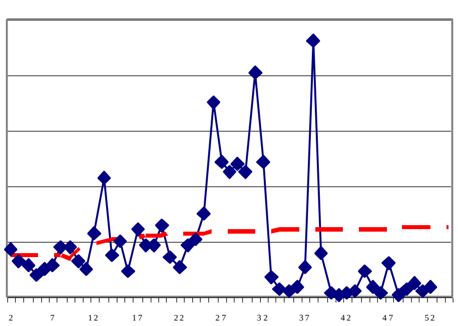


Actual and fitted in year 5

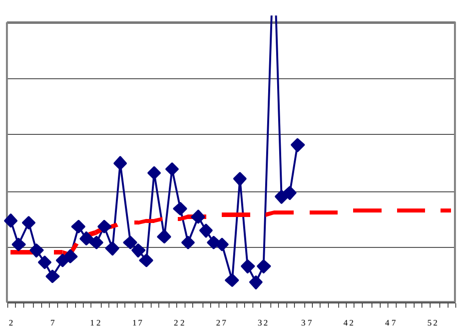


Payment factor submodel

Actual and fitted in year 1



Actual and fitted in year 5



The robot in action

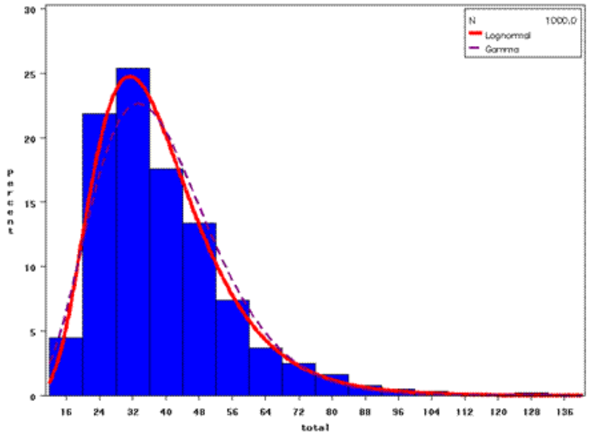
- Results by model

Accident	PPCI		PPCF		PCE	
year	Mean	CV	Mean	CV	Mean	CV
1	8	229%	132	55%	22	105%
2	20	216%	242	47%	56	108%
3	58	166%	165	58%	23	98%
4	110	135%	268	47%	70	90%
5	242	100%	861	30%	317	62%
6	292	71%	1,216	27%	671	64%
7	680	59%	1,257	27%	799	44%
8	819	53%	1,672	27%	1,319	40%
9	2,262	49%	3,366	25%	2,040	32%
10	3,546	49%	3,510	22%	2,368	31%
11	6,363	48%	6,041	21%	5,480	31%
12	7,151	46%	6,742	20%	6,700	31%
13	8,461	44%	8,664	21%	7,234	33%
14	8,904	42%	9,015	21%	3,749	98%
Total ex 14	30,011		34,136		27,099	
Total	48,589	42%	41,721	18%	29,366	22%

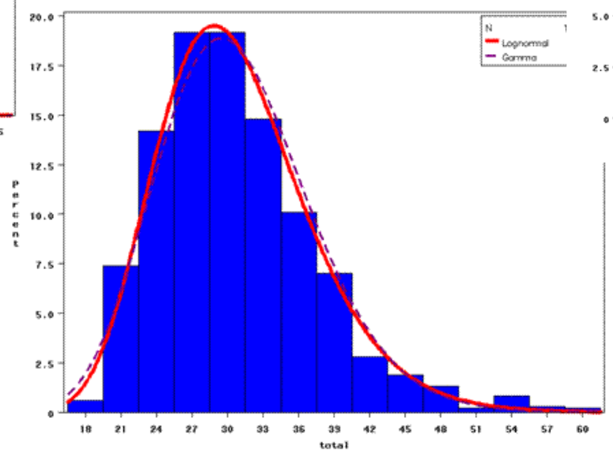
The robot in action

- Distribution of results [from bootstrap]

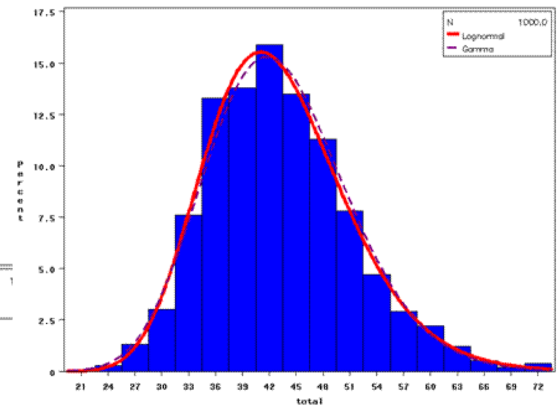
PPCI



PCE



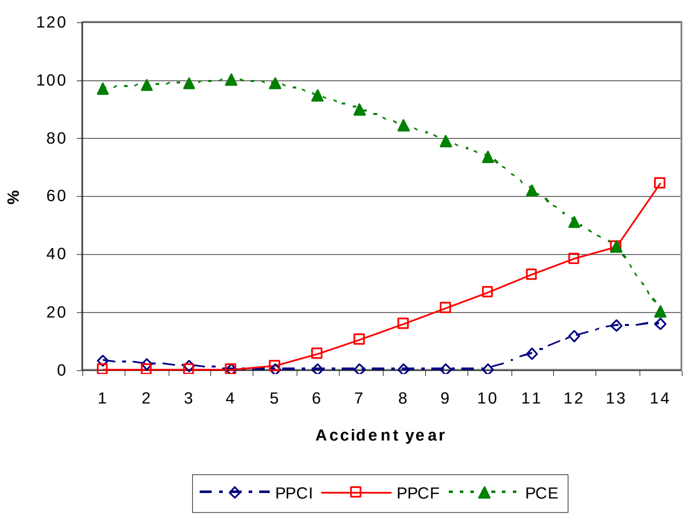
PPCF



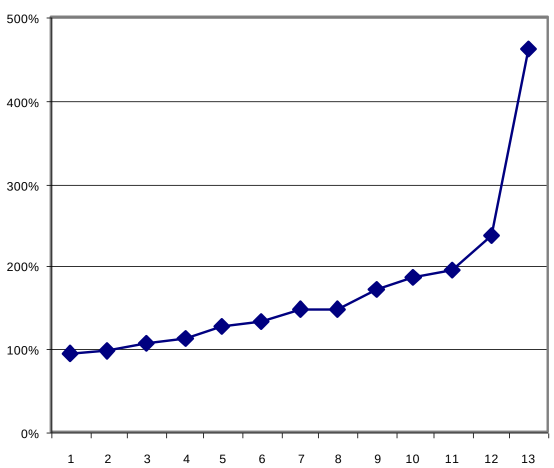
The robot in action

- Blending

Smoothness of weights



Smoothness of liability/case est



The robot in action

- Blended results

Accident	L=0	
year	Mean	CV
1	22	104%
2	56	107%
3	24	96%
4	70	90%
5	324	60%
6	702	58%
7	847	38%
8	1,375	32%
9	2,317	24%
10	2,672	21%
11	5,712	20%
12	6,771	18%
13	8,035	17%
14	7,963	20%
Total	36,891	12.7%

Discussion

- Process for automating large parts of valuation
  - Push a button to get liability estimates and coefficients of variation
  - Diagnostics warn when models fitting poorly
  - Of course can't always be used
    - Eg big changes in experience
  - Potential to save a lot of time where regular valuations carried out

References

- De Jong, P. and Zehnwirth, B. (1983). Claims reserving state space models and the Kalman filter. Journal of the Institute of Actuaries, 110, 157-181.
- Kalman, R.E. (1960). A new approach to linear filtering and prediction problems. Journal of Basic Engineering, 82, 340-345.
- McGuire, G. and Taylor, G. (2007). Adaptive Reserving using Bayesian revision for the Exponential Dispersion Family. Taylor Fry working paper.
- Stoffer, D.S. and Wall, K.D. (1991). Bootstrapping state-space models: Gaussian maximum likelihood estimation and the Kalman Filter. Journal of the American Statistical Association, 86 (416), 1024:1033.
- Taylor, G. (2000). Loss reserving: an actuarial perspective. Kluwer Academic Publishers. London, New York, Dordrecht.
- Taylor, G. (2008). Second order Bayesian revision of a generalised linear model. Scandinavian Actuarial Journal, in press.