Unbiased Development for Individual Claims—Taming the Wild Burning Cost

Joseph Boor, FCAS, Ph.D., CERA

Actuary

The Florida Office of Insurance Regulation

Presentation to 2017 Casualty Actuarial Society Reinsurance Seminar

June 5-6, 2017

Antitrust Notice

- The Casualty Actuarial Society is committed to adhering strictly to the letter and spirit of the antitrust laws. Seminars conducted under the auspices of the CAS are designed solely to provide a forum for the expression of various points of view on topics described in the programs or agendas for such meetings.

- Under no circumstances shall CAS seminars be used as a means for competing companies or firms to reach any understanding expressed or implied that restricts competition or in any way impairs the ability of members to exercise independent business judgment regarding matters affecting competition.

- It is the responsibility of all seminar participants to be aware of antitrust regulations, to prevent any written or verbal discussions that appear to violate these laws, and to adhere in every respect to the CAS antitrust compliance policy.

What is Burning Cost (for Excess)?

Key Features

- Fairly early on in accident year
- Reserving an excess layer, but have ground up loss values
- Not a lot of excess losses to make excess loss triangle/dont rely on excess triangle
- Multiply each ground up loss by average LDF (for this AY/RY).
- See which losses pierce excess retention, and by how much, to get “ultimate” excess loss

What is Wrong with Burning Cost?

- Claims do not all develop by the same percentage-some more, some less
  - Logical that claims that eventually become large develop more than average
  - Differences in actual development from claim to claim more consistent with a probability distribution for development
- IBNR not included
  - Discussed later

Conceptual Correction of Burning Cost—Setup

- Have random variables "X" and "Y" for
  - The random amount (severity) of an individual undeveloped loss ($\underline{X}$)
  - The random amount of an individual ultimate loss ($\underline{Y}$)
- Add in a random development factor $\underline{R}$ so that, for actual claim values x, r, y, we know $x \times r = y$

Conceptual Correction of Burning Cost—Eliminating Bias

- Require $X \times R \sim Y$
- This means that $X \times R$ and Y are equal in distribution
- That means that a random sample from X multiplied by a random sample from R is, a priori, equal to a random sample from the entire Y distribution
  - Thus, the expected value of $X \times R$ in any layer is the expected value of Y in the layer
  - $X \times R$ generates unbiased estimates of the ultimate losses, Y's, in any layer

Turnkey Methodology for Estimating the Probability Mass Function $s_R(r)$ of R

All the methods may be implemented using the standard spreadsheet software on my computer

... performing one method is challenging, though.

Need a lot of methods for different situations—will abbreviate some items to stay within time limit.

Will just show math—ask questions if need advice on doing computations/spreadsheet implementation.

Log Transform

- Finding a multiplier R such that $X \times R \sim Y$ is really hard

- Finding an additive distribution is easier, so take logs to get $\ln(X) + \ln(R) \sim \ln(Y)$

- Simplify the symbols with random variables $\underline{U} = \ln(X)$, $\underline{Z} = \ln(R)$, $\underline{W} = \ln(Y)$, $U + Z \sim W$

Formulas to Convert Initial Severity Distributions to those of Log Distributions

For example, s_U is a probability distribution like s_X, and must total 1.0. Need to use formula (divide by derivative of transform) for substitution of variables $\frac{du}{dx}dx$ in integral from basic calculus for densities of U, Z, W.

$$\begin{aligned}
\frac{s_U(\ln(x))}{x} &= s_X(x) = s_X(\exp(u)); s_X(\exp(u))\exp(u) = s_U(u); s_X(\exp(u)) = s_U(u)\exp(-u); \\
\frac{s_Z(\ln(r))}{r} &= s_R(r) = s_R(\exp(z)); s_R(\exp(z))\exp(z) = s_Z(z); s_R(\exp(z)) = s_Z(z)\exp(-z); \text{ and,} \\
\frac{s_W(\ln(y))}{y} &= s_Y(y) = s_Y(\exp(w)); s_Y(\exp(w))\exp(w) = s_W(w); s_Y(\exp(w)) = s_W(w)\exp(-w).
\end{aligned}$$

The Matrix Method

First Basic Method —The Matrix Method

- For start
  - Take sets of points "g" apart, in U, Z, and W.
  - Assign approximate probability in each interval to the discrete point representing the interval
    * $[\mathcal{U}]_i = gs_U(ig) = gs_X(\text{exp}(ig))\,\text{exp}(ig)$'s for $\underline{i} = 0, 1, 2, ..., l$
    * $[\mathcal{Z}]_j = gs_Z(jg) = gs_R(\text{exp}(jg))\,\text{exp}(jg)$'s for $\underline{j} = 0, 1, 2, ..., m$
    * $[\mathcal{W}]_k = gs_W(kg) = gs_Y(\text{exp}(kg))\,\text{exp}(kg)$'s for $\underline{k} = 0, 1, 2, ..., n$

Key Start to Matrix Method

- Assumption on last slide that the indices $i,\ j,\ k$ start at zero is not required, and is not always best approach
  - But is best for the illustration

- Core of this method=how can the index i of U and the index j for Z add to zero for index k of W? U and Z must both be zero

- So, up to effects of using discrete points, $[\mathcal{W}]_0 \approx [\mathcal{U}]_0 \times [\mathcal{Z}]_0$

Key Start to Matrix Method

- Similarly, for $W = 1$, one index of U or Z must be one, the other must be zero $[\mathcal{W}]_1 \approx [\mathcal{U}]_0 \times [\mathcal{Z}]_1 + [\mathcal{U}]_1 \times [\mathcal{Z}]_0$

- Continuing the process, we get

$$[\mathcal{W}]_k \approx \sum_{i=0}^{k} [\mathcal{U}]_i \times [\mathcal{Z}]_{k-i}.$$

The Matrix Equation

- Result is matrix equation $[\mathcal{W}] = [\mathcal{U}^*] \times [\mathcal{Z}]$, where $[\mathcal{U}^*]$ is

$$[\mathcal{U}^*] = \begin{bmatrix} \mathcal{U}_0 & 0 & 0 & ... \\ \mathcal{U}_1 & \mathcal{U}_0 & 0 & ... \\ \mathcal{U}_2 & \mathcal{U}_1 & \mathcal{U}_0 & ... \\ ... & ... & ... & ... \end{bmatrix}$$

  - (brackets dropped inside the matrix).

- Note that the indices need not start at zero, so $[\mathcal{U}^*]$ could have a different shape, but they must be subject to the same additive principles

Solution—Estimate of R

- Matrix equation may be overconstrained ($i > j$), so have best estimate $[\mathcal{Z}] = \left([\mathcal{U}^*]^T \times [\mathcal{U}^*]\right)^{-1} \times \left([\mathcal{U}^*]^T \times [\mathcal{W}]\right)$

- Then, for the $r_j = \exp(jg)$'s, $s_R(r_j) = [\mathcal{Z}]_j/(g \times \exp(jg))$ (point estimate-for curve fit)

- Could use $[\mathcal{Z}]_j$'s as weights for development factor r_j's (be sure development into excess layer is covered)

- All the matrix setup and equation solution may be done using standard spreadsheet software

Matrix Method Example

Step 1–Calculation of $[\mathcal{U}]$ from Values of s_X

First step... (grid spacing $=g=.3$) (all input data assumed)

u (or $.3i$)	$x = \exp(u)$	$s_X(x)$	$s_U(u) = x s_X(x)$	$[\mathcal{U}]_i = .3 s_U(u)$
0	1.000	.333	.333	.100
.3	1.350	.494	.667	.200
.6	1.822	.549	1.000	.300
.9	2.460	.339	.833	.250
1.2	3.320	.151	.500	.150

Step 2–Calculate the $[\mathcal{W}]$ for various indices k from values of s_Y

Step 3–Matrix Equation

$$[\mathcal{U}*] \times [\mathcal{Z}] = [\mathcal{W}], \text{ or}$$

$$\begin{bmatrix} .10 & 0 & 0 & 0 \\ .20 & .10 & 0 & 0 \\ .30 & .20 & .10 & 0 \\ .25 & .30 & .20 & .10 \\ .15 & .25 & .30 & .20 \end{bmatrix} \times [\mathcal{Z}] = \begin{bmatrix} .010 \\ .040 \\ .100 \\ .185 \\ .235 \end{bmatrix}$$

Step 4–Matrix Algebra Spreadsheet Program Best Estimate Solution

Solution fulfills $[\mathcal{U}^*]^T \times [\mathcal{U}^*] \times [\mathcal{Z}] = [\mathcal{U}^*]^T \times [\mathcal{W}]$ or

$$\begin{bmatrix} .2250 & .1925 & .1250 & .0550 \\ .1925 & .2025 & .1550 & .0800 \\ .1250 & .1550 & .1400 & .0800 \\ .0550 & .0800 & .0800 & .0500 \end{bmatrix} \times [\mathcal{Z}] = \begin{bmatrix} .12050 \\ .13825 \\ .11750 \\ .06550 \end{bmatrix}$$

Step 5–Results of Final Square Matrix Algebra (from Spreadsheet Program)

$$[\mathcal{Z}] = \begin{bmatrix} .1 \\ .2 \\ .3 \\ .4 \end{bmatrix}$$

Step 6 –Results of Using Discrete Random Development Factors

	Index "j"				$[\mathcal{Z}]_j$ Wtd.
	0	1	2	3	Average
$[\mathcal{Z}]_j$	0.1	0.2	0.3	0.4	
$\exp(.3j) = r = LDF$	1.000	1.350	1.822	2.460	
Loss 1	$5,000	$5,000	$5,000	$5,000	
Developed	$5,000	$6,749	$ 9,111	$12,298	
Excess $100,000	$0	$0	$0	$0	$0
Loss 2	$50,000	$50,000	$50,000	$50,000	
Developed	$50,000	$67,493	$91,106	$122,980	
Excess $100,000	$0	$0	$0	$22,980	$9,192
Loss 3	$75,000	$75,000	$75,000	$75,000	
Developed	$75,000	$101,239	$136,659	$184,470	
Excess $100,000	$0	$1,239	$36,659	$84,470	$45,034
Total Est. Excess					$54,226

Step 7 –Results of Using Curve Fit Random Development Factors (Poor Fit)

	Index "j" 0	1	2	3
$[\mathcal{Z}]_j$	0.1	0.2	0.3	0.4
$\exp(.3j) = r$	1.000	1.350	1.822	2.460
$s_R(r) = [\mathcal{Z}]_j/(.3\exp(.3j))$	0.33333	0.49387	0.54881	0.54209
Uniform distribution of Best Fit: Avg. Value s = .47953; Inverse=Interval Length = 2.0837 (Use 2.0) s_R-Wtd. Avg. of Points = Center of Interval=1.7378 (Use 1.7) Selected Uniform Distribution with Mass .5 on $[.7, 2.7)$ Mahler Excess Function = $\int_{100{,}000/C}^{2.7} .5(rC - 100,000)dr$, for each claim amount C such that $1.7C \geq 100,000$				
Loss 1	5,000	Excess =	0	
Loss 2	50,000	Excess =	6,125	
Loss 3	75,000	Excess =	35,021	
Total			41,146	

Matrix Method Enhancements

Some Possible Basic Improvements in the Matrix Method (More in Paper)

- "Twice" as many rows (i's) as columns (j's)
- Correct mean
- Correct variance
- Correct total probability

Another Possible Basic Improvement in the Matrix Method

- Instead of starting near zero, focus on the upper end of the distribution
  - Also to target LDFs most likely to generate excess claims

$$[\mathcal{U}^*] = \begin{bmatrix} ... & ... & ... & ... \\ ... & ... & ... & ... \\ ... & \mathcal{U}_l & \mathcal{U}_{l-1} & \mathcal{U}_{l-2} \\ ... & 0 & \mathcal{U}_l & \mathcal{U}_{l-1} \\ ... & 0 & 0 & \mathcal{U}_l \end{bmatrix},$$

Curve Fitting Methods

Fitting Z via Mean and Variance Matching

- We already know
  - Mean of Z is $E[Z] = E[W] - E[U]$, W and U are known.
  - Variance of Z is $Var[Z] = Var[W] - Var[U]$.
- Can use method of moments to fit Pareto, etc. distibution
- Important to choose family of distributions that has approximately right large loss potential.

Fitting a Distribution for Z by Matrix-Based Parameter Estimation

- Method:
  - Pick curve family
  - Pick smallish number of points (j's) on which to compute $[\mathcal{Z}]_j$'s using current selected curve
  - Compute $[\mathcal{U}^*]$ corresponding to i's, k's, U
  - Pick initial values determining curve (Step 4)
  - Multiply $[\mathcal{Z}]$ by determined $[\mathcal{U}^*]$ and compare to $[\mathcal{W}]$ (sum of squared errors, etc.)
  - Have spreadsheet program change values determining curve and go to Step 4 until best estimate found

Example of Matrix-Based Parameter Estimation

This method best illustrated by example...

Optimal Pareto Parameters	$x_M =$			2.79		$\alpha =$	1.64	
Index	$[\mathcal{U}^*]$				Pareto Values $[\mathcal{Z}]$	$[\mathcal{U}^*] \times [\mathcal{Z}]$	$[\mathcal{W}]$	Squared Error	Weight
0	0.1	0.0	0.0	0.0	0.2618	0.0262	0.010	0.00026	4
1	0.2	0.1	0.0	0.0	0.1411	0.0665	0.040	0.00070	5
2	0.3	0.2	0.1	0.0	0.0855	0.1153	0.100	0.00023	6
3	0.4	0.3	0.2	0.1	0.0562	0.1698	0.185	0.00023	7
4	0.5	0.4	0.3	0.2		0.2243	0.235	0.00003	8
							Weighted Sum =	.0078	

Matching Pareto Parameters of Y

- Sometimes, very (or mostly) upper layer losses are targeted
- Pareto is oft-used in this layer
- Paper shows (Penderzoli and Rathie, probability of sum of Pareto distributions), that when Y has Pareto character with shape parameter α, so does R
- May compute Pareto parameter with probabilities/percentiles p_1, p_2 near unity and cumulative severity distribution F_Y of Y

$$\alpha = \frac{\ln\left(\frac{1-p_1}{1-p_2}\right)}{\ln\left(\frac{F_Y^{-1}(p_2)}{F_Y^{-1}(p_1)}\right)}.$$

Fourier Analysis—A Heavily Mathematical Approach

- Fourier transform (in this case, characteristic function) changes a random variable X to a separate function φ_T, with a separate independent variable (ω), i.e. $\varphi_T(\omega) = E[\exp(i\omega X)]$

- Nice property $\varphi_U(\omega) \times \varphi_Z(\omega) = \varphi_W(\omega)$,
  or $\varphi_Z(\omega) = \varphi_W(\omega)/\varphi_U(\omega)$ (for all ω)

- Are you prepared to explain that the "i" part gives you an "imaginary" number

- My spreadsheet software has a discrete Fourier transform, but it is poorly documented-I referred to this earlier

Testing the Results

- Helpful to take X and the computed R and run Monte Carlo simulation of Y

- Put careful attention on the layer you are targeting.

- Especially if the first approach misses Y considerably, consider using more than one method.

Finding External Data for X and Y and Making the Most of It.

Reason for Using External Data

- If develop R off data X and Y that are from the dataset to be developed , then you'll always just get Y

- may work if Y is from prior years in fully, fully credible (including upper layers) program

Sources of External Data - Internal to Company

- Distributions from Larger Bodies of Claims

- Have countrywide distribution stand in for state data

- Total (all programs combined) or larger program data for individual program.

- With adjustment formulas on next page, may reasonably correct data with different claims claims handling, different close by maturities, etc.

Modify Mean and Variance to Match Patterns of Baseline Data

- May have, e.g., TPA-handled program when most data has in-house handling
  - Have adjustment factor $\frac{LDF_{alternate}}{LDF_{benchmark}} \times X$ for average/mean LDF difference
  - For variance, could transform x to

$$x_{transformed} = \mu_X + \frac{\beta}{\alpha}(x - \mu_X),$$

$$\beta = \text{S.D. of Benchmark}, \ \alpha = \text{S.D. of specific data.}$$

    * Makes variance look like variance of benchmark distribution, then apply R.
    * Better approach using geometric mean/variance characteristics in paper.

Advisory Organization Data

- Can estimate ultimate severity from ILF's/ELPPF's
- Use various circulars creatively
- Consider purchasing data.

IBNR Claims

Pure IBNR Claims

- Potential Issues with IBNR Claims
  - They don't get included when you develop individual claims
  - They may tend to be larger than the claims reported to date

Resolution of IBNR Issue

- IBNR claims may be larger, but the ultimate loss distribution s_Y accounts for all claims, so claims developed by the random development factor R reflect the costs of all claims, even the IBNR claims not even present in X.

- Do need to multiply each final excess cost computation (not just some property of Y) by count development factor.

Summary

- Wide variety of methods and proposals for source data for random development factors presented
- Should make the process a reasonable option for most practitioners.

???