Bayesian Claim Severity with Mixed Distributions

by Benedict Escoto

ABSTRACT

This paper presents a Bayesian technique for adjusting a mixed exponential severity distribution in response to partially-credible observed claim severities. It presents two applications: pricing excess of loss (XOL) reinsurance layers and computing increased limits factors (ILFs). The paper's Bayesian model uses a Dirichlet distribution over the mixed exponential's initial mixture weights. The posterior distribution, produced by conditionalizing on the observed claim severities, is computed using a Markov chain Monte Carlo method.

KEYWORDS

Bayesian, mixed exponential distribution, severity, credibility, Dirichlet

1. Introduction

Actuaries find per-claim loss severity distributions useful for several purposes, including computing increased limit factors (ILFs), pricing excess of loss (XOL) reinsurance contracts, and stochastic risk modeling. This paper presents a method for computing a severity distribution in the mixed exponential family.

When claim data is abundant, a severity distribution can be easily computed using empirical methods, nonparametric statistics, or maximum likelihood methods (Klugman 2009). When no data is available, actuaries use a default distribution, perhaps from a statistical agency or from a different book of business. Our technique is designed to compute a severity distribution for the awkward middle ground where data is partially credible: some claims have been observed, but the default distribution is still considered informative.

Intuitively, the claim severity distribution should morph from the default distribution into some fitted distribution as more and more claims are observed. This paper uses Bayesian statistics to achieve this "morphing" in a principled and mathematically correct way.

Here are some advantages of the technique presented in this paper:

1. The default severity distribution is mixed exponential and thus is compatible with many severity distributions supplied by Insurance Services Office (ISO). Other mixed distributions should work with minor modifications.
2. In the limiting condition where there are no claims, the method produces the default distribution—having too few claims is never an issue.
3. In the other limiting condition (abundant observed claims), the method is equivalent to fitting a mixed exponential via maximum likelihood.
4. The uncertainty about the default severity distribution is supplied by a single parameter, which can be estimated in a straightforward way.
5. Truncated and censored data (as caused by limits and deductibles) are handled correctly. Adjustments such as the Kaplan-Meier estimator are not required.
6. The posterior distribution is computed using standard Bayesian updating on observed claim severities. This is usually considered the most principled way of analyzing partially-credible data.
7. Trend can be a chicken-and-egg problem: claim severity depends on trend, but the estimation of trend depends on the credibility of observed claim severities. Here trend and severity are estimated simultaneously.
8. The technique quantifies uncertainty about the severity distribution. In other words, it analyzes "parameter risk" and not just "process risk."
9. The resulting posterior predictive severity distribution is easy to work with analytically because it is a mixed exponential with the same means as the default severity distribution.

The numerical results of this technique can be computed quickly using the Markov chain Monte Carlo (MCMC) method. Below we will use this method on some sample data to compute the posterior distribution. Later the paper discusses two possible applications: pricing excess of loss reinsurance and generating ILFs.

2. Required input data

We need four pieces of input data to use the technique to compute a posterior severity distribution:

1. the observed claim data: a per-claim list of severity, age, deductible, and whether or not each claim was capped by a policy limit,
2. the means and weights of the default mixed exponential severity distribution,
3. the prior severity uncertainty, as represented by Dirichlet parameter α_0, and
4. the prior trend mean and standard deviation.

The first, claim data, is straightforward. The other inputs are more subjective. The remainder of this section discusses each of these and gives some examples.

2.1. Claim data

Table 1 shows the format of the necessary claim data. The Amount column contains the total claim size (this could include or exclude loss expense, depending on what's being modeled). In this example, we are using closed claims only, so no development is necessary. If development is required, claims should be developed prior to the application of this method.

The Age column is used for trending and indicates years before the desired trend endpoint. Fractional ages are allowed. The capped column indicates whether the claim was capped by policy limits. The deductible column indicates the deductible that the claim was subjected to. If the deductibles were non-zero, the indicated amount would be the loss net of the deductible.

2.2. Mixed exponential

The second piece of data required by this method is the mixed exponential severity distribution that would be used if no claim data were available. Here we use a Bayesian framework, where it is called the prior predictive distribution. Table 2 shows the means and weights of an example mixed exponential distribution. Common sources for this data are statistical organizations such as ISO or analysis of a similar book of business. For weights a_j and means μ_j, the probability density function of a mixed exponential is

$$f(x) = \sum_{j=1}^{m} \frac{a_j}{\mu_j} e^{-\frac{x}{\mu_j}}.$$

Table 1. Observed claim data

Amount	Age	Deductible	Capped?
33,750	3	0	No
1,000,000	1	0	Yes
22,707	1	0	No
54,135	1	0	No
174,524	3	0	No
19,661	2	0	No
140,735	2	0	No
1,000,000	3	0	Yes
1,127	1	0	No
316,483	2	0	No

Table 2. Prior means and weights

Weights (%)	Means
30.0	50,000
25.0	100,000
25.0	500,00
10.0	1,500,000
7.0	5,000,000
3.0	20,000,000
Avg	1,265,000

This paper focuses on the mixed exponential distribution because that is the prior severity distribution most familiar to the author. However, this paper's techniques should apply to mixtures of other distributions (e.g., an Erlang mixture as developed by Lee and Lin 2010) with modifications. The main computational requirement is that the distribution's density and cumulative distribution function should be calculable.

2.3. Dirichlet uncertainty

Although the means and weights of the mixed exponential distribution determine the severity distribution, we also need to know how certain we are that these parameters are accurate. Are they just a rough estimate that may be far off, or are we sure that the true distribution is very similar? The more confidence we have in our a priori distribution, the longer we will stick with it as more claims are observed. This uncertainty can be summarized as α_0, the sum of the Dirichlet parameters. Here we picked a value of 20. See Section 3.3 for guidance on choosing this parameter.

2.4. Trend

Finally, we need the prior trend rate distribution. The trend is applied to each ground-up claim by age. For instance, if the trend rate is 7%, then claims of age 2.5 are expected to be $1.07^{-2.5}$ times as severe as claims of age 0.

We assume trend is constant over time, and the parameter uncertainty is modeled as a gamma distribution, shifted so that the zero point indicates −100% trend. Choosing a mean and standard deviation suffices to determine the gamma parameters via method of moments. These constants can be based on a statistical agency, another book of business, economic indicators, or actuarial judgment. In this paper, the example trend mean is 0.05 and the trend standard deviation is 0.01.

3. Bayesian model

The expressions below formally describe the assumed Bayesian model. See Hoff (2009) for a basic introduction to Bayesian methods.

$$x_i|b_i,r \sim \min\left(\text{Exponential}\left(\frac{r^{t_i}}{\mu_{b_i}}\right), l_i\right). \tag{1}$$

$$b_i|w_1,\ldots,w_m \sim \text{Categorical}(g_{d_i}(w_1),\ldots,g_{d_i}(w_m)). \tag{2}$$

$$w_1,\ldots,w_m \sim \text{Dirichlet}(\alpha_1,\ldots,\alpha_m). \tag{3}$$

$$r \sim \text{Gamma}(k,\theta). \tag{4}$$

x_i, b_i, w_b, and r are all real random variables, while μ_b, m, t_i, d_i, l_i, α_j, k, and θ are real numbers. Index $i \in \{1, \ldots, n\}$ ranges over the number of observed claims n; x_i are the claims' severities, l_i are their limits, and d_i are their deductibles. m is the number of mixed exponential buckets; $b_i \in \{1, \ldots, m\}$ represents the bucket that produced the ith claim, and μ_b is the mean of each bucket b. Functions g_{d_i} adjust the weights for each claim's deductible d_i; they are defined by Equation (24) in Section 5. Finally, t_i is the age of the ith claim and r is the trend factor. Random variables are independent unless the dependence is expressed in the equations above.

Intuitively, Equations (1) and (2) express the process risk and (3) and (4) express the parameter risk. They basically say that each claim is independent and is produced by a (possibly capped) mixed exponential distribution. Moreover, we know for certain the exponential means $\mu_1, \ldots, \mu_m$. However, the trend rate and mixed exponential weights are uncertain. All claims share the same trend factor r and mixed exponential weights $w_1, \ldots, w_m$ but these are themselves random variables, generated by the gamma and Dirichlet distributions. Sometimes α_j, k, and θ are called hyperparameters because they generate parameters (w_b and r) used in other distributions.

3.1. The categorical distribution

Actuaries know and use the categorical distribution but perhaps do not bother naming it. This distribution simply chooses a number from 1 to m with specified probabilities. For instance, if $y \sim$ Categorical (.5, .3, .2) then $y = 1$ with probability 0.5, $y = 2$ with probability 0.3, and $y = 3$ otherwise. Equations (1) and (2) imply that x_i is a capped and trended mixed exponential distribution, conditional on weights $w_1, \ldots, w_m$.

3.2. The Dirichlet distribution

The exponential, categorical, and gamma distributions are familiar to actuaries, the Dirichlet perhaps less so. The Dirichlet distribution is the multidimensional analogue of the beta distribution. Just as the beta distribution can be used to express uncertainty about two numbers that must sum to 1, the Dirichlet can express uncertainty about m positive numbers that must sum to 1. And just as the beta distribution can parameterize a Bernoulli distribution, the Dirichlet is natural for parameterizing a categorical distribution. This property makes it popular in Bayesian analysis (see Mildenhall 2006, for an example of the Dirichlet applied in an insurance context).

Here are some useful properties of the Dirichlet distribution (Wikipedia 2010). If $w_1, \ldots, w_m \sim$ Dirichlet $(\alpha_1, \ldots, \alpha_m)$ and $\alpha_0 = \sum_{j=1}^m \alpha_j$, then

$$\textstyle\sum_{j=1}^m w_j = 1. \tag{5}$$

$$E[w_j] = \frac{\alpha_j}{\alpha_0}. \tag{6}$$

$$\text{Var}[w_j] = \frac{\alpha_j(\alpha_0-\alpha_j)}{\alpha_0^2(\alpha_0+1)} = \frac{E[w_j](1-E[w_j])}{(\alpha_0+1)}. \tag{7}$$

$$\text{Cov}[w_j,w_k] = \frac{-\alpha_j\alpha_k}{\alpha_0^2(\alpha_0+1)} = \frac{-E[w_j]E[w_k]}{\alpha_0+1} \quad \text{for } j \neq k. \tag{8}$$

Because we are given the initial default weights $a_j = \mathrm{E}[w_j]$, the choice of α_0 will uniquely determine $\alpha_1, \ldots, \alpha_m$ by (6). As in the beta distribution, the larger the sum of the parameters α_0, the more certain we are of the true weights.

3.3. Choosing α_0

In the input data section, the parameter α_0 was used to summarize our uncertainty around the mixed exponential prior distribution. Sometimes this parameter is thought of intuitively as the number of claim observations equivalent to the information encapsulated by our Bayesian prior. In this case, however, that intuition is of limited help in selecting a particular value. Instead we will use this result:

Suppose that X is a mixed exponential distribution (with fixed means) depending on weights $w_1, \ldots, w_m$, and that $h(x)$ is a real function. Define $h_j = \mathrm{E}[h(x)|b = j]$ and $a_j = \dfrac{\alpha_j}{\alpha_0}$. Then

$$\mathrm{Var}[E[h(x)|w_1,\ldots,w_m]] = \sigma^2$$

$$= \mathrm{Var}\left[\sum_{j=1}^{m} w_j E[h(x)|b=j]\right]$$

$$= \mathrm{Var}\left[\sum_{j=1}^{m} w_j h_j\right] \tag{9}$$

$$= \sum_{j=1}^{m} h_j^2 \mathrm{Var}[w_j] + \sum_{j\neq k} h_j h_k \mathrm{Cov}[w_j, w_k] \tag{10}$$

$$= \sum\nolimits_{j=1}^{m} h_j^2 \frac{a_j(1-a_j)}{\alpha_0+1} + \sum\nolimits_{j\neq k} h_j h_k \frac{-a_j a_k}{\alpha_0+1}, \tag{11}$$

hence

$$\alpha_0 = \sigma^{-2}\left(\sum_{j=1}^{m} h_j^2 a_j(1-a_j) - \sum_{j\neq k} h_j h_k a_j a_k\right) - 1. \tag{12}$$

Above, (9) follows from (1) and (2), (10) from the variance sum formula, and (11) from (7) and (8).

We can now use (12) to choose a value for α_0. For instance, suppose the actuary feels that the there is about \$50,000 of uncertainty around the expected value of the actual ground-up mixed exponential claim distribution capped at \$1M, where uncertainty is measured by the standard deviation. Then, by setting $h(X) = \min(X, \text{1Mil})$, $h_j = \mu_j\left(1 - e^{-\frac{1Mil}{\mu_j}}\right)$, and $\sigma =$ \$50,000, the above result can be used to calculate α_0.

In our sample input data, we assumed that the standard deviation of expected loss in the first million layer was 65770. Applying (12), $\alpha_0 = 20$.

3.3.1. Other metrics

Although (12) gives us a quick way to estimate α_0 from prior uncertainty about the standard deviation of expected loss, the actuary's prior may be couched in different terms. For instance, he or she may instead have an intuition about a 90% confidence interval around expected loss, or even a 90% confidence interval around the 95% percentile of loss.

A simple simulation can be used to infer α_0 from this type of information:

1. Using a candidate value of α_0, simulate 500 draws from the Dirichlet distribution.
2. For each of the draws in (1), simulate 1000 claims, applying deductibles or limits as desired.
3. For each draw in (1), compute the statistic (e.g., expected loss) on the 1000 claims generated in (2).
4. Compute the summary statistic (e.g., standard deviation) on the 500 statistics computed in (3).

Modern computers are fast enough that the above procedure may just take a split second per α_0 value. It is thus practical to evaluate a range of values and choose one based on an arbitrary metric.

3.4. Matching the prior predictive severity distribution

In the introduction, we mentioned that the model matches the prior predictive mixed exponential severity distribution before any claims are observed. This is demonstrated below, by showing that our model's CDF is equal to the CDF of the desired default mixed exponential. For simplicity we will assume here that there are no limits or deductibles—if two distributions are the same gross of limits and deductibles, they will also be the same net of those policy terms.

$$F(c) = \text{Prob}[x_i < c] \tag{13}$$

$$= \sum_{j=1}^{m} \text{Prob}[x_i < c | b_i = j] \text{Prob}[b_i = j] \tag{14}$$

$$= \sum_{j=1}^{m} \text{Prob}[x_i < c | b_i = j]$$

$$\int \text{Prob}[b_i = j | w_j] f(w_1, \ldots, w_m) dw \tag{15}$$

$$= \sum_{j=1}^{m} \text{Prob}[x_i < c | b_i = j] E[w_j] \tag{16}$$

$$= \sum_{j=1}^{m} \left(1 - e^{-\frac{cd^{t_i}}{m_j}}\right) a_j \tag{17}$$

$$= \sum_{j=1}^{m} a_j \left(1 - e^{-\frac{c}{m_j}}\right) \tag{18}$$

$$= 1 - \sum_{j=1}^{m} a_j e^{-\frac{c}{m_j}}. \tag{19}$$

Steps (13) to (17) follow from simple probability and the model specification. Step (18) holds because our default distribution is taken to cover the current time, so $t_i = 0$. Finally, $\sum_{j=1}^{m} a_j = 1$ implies (19), the CDF of our default mixed exponential distribution.

3.5. Deductibles

This subsection will describe how this method accounts for deductibles and explain how the *g* functions mentioned in Section 3 are derived. The main result will be that a mixed exponential ground-up severity distribution implies that losses net of any deductible level are also mixed exponential with the same means.

Suppose a claim has a mixed exponential ground-up severity distribution represented by random variable *X*. Then its probability density function is

$$f_X(x) = \sum_{j=1}^{m} \frac{a_j}{\mu_j} e^{-\frac{x}{\mu_j}}$$

for means μ_j and weights a_j. The loss net of deductible *d* is distributed as $Y = (X - d)|\{X >= d\}$. Its density function is

$$f_Y(x) = \frac{f_X(x+d)}{1 - F_X(d)} \tag{20}$$

$$= \frac{\sum_{j=1}^{m} \frac{a_j}{\mu_j} e^{-\frac{x+d}{\mu_j}}}{\sum_{j=1}^{m} a_j e^{-\frac{d}{\mu_j}}} \tag{21}$$

$$= \frac{\sum_{j=1}^{m} \frac{a_j e^{-d/\mu_j}}{\mu_j} e^{-\frac{x}{\mu_j}}}{\sum_{j=1}^{m} a_j e^{-\frac{d}{\mu_j}}} \tag{22}$$

$$= \sum_{j=1}^{m} \frac{a'_j}{\mu_j} e^{-\frac{x}{\mu_j}}, \tag{23}$$

and therefore the new distribution is also mixed exponential with means μ_j and new weights a'_j where $a'_j = \frac{a_j e^{-\frac{d}{\mu_j}}}{\sum_{k=1}^{m} a_k e^{-\frac{d}{\mu_k}}}$.

By defining

$$g_d(w_j) = \frac{w_j e^{-\frac{d}{\mu_j}}}{\sum_{k=1}^{m} w_k e^{-\frac{d}{\mu_k}}} \tag{24}$$

we adjust the mixed exponential weights to reflect the deductible *d*. From this definition we can make two observations:

1. If there is no deductible, then $g_0(w_j) = \frac{w_j \times 1}{\sum_{k=1}^{m} w_k \times 1}$ $= w_j$. Thus, we can simply ignore the *g* functions in this case.
2. The larger the deductible *d,* the more the weights of the mixed exponential are skewed to the larger means by the $e^{-\frac{d}{\mu_j}}$ term. This is intuitive because exponentials with larger means tend to generate more severe claims.

This method correctly adjusts each claim's severity distribution to account for its deductible. However, it suffers from two limitations when claims are subject to different deductible levels. First, it assumes that claims subject to different deductible levels are generated from the same underlying ground-up distribution. In reality, policyholders' perceptions of their risk influence their deductible choices. Second, it only takes into account per-claim severity data. Frequency data along with exposure information per deductible level (e.g., a deductible profile) may yield additional information about ground-up severity.

3.6. Why not vary the means?

It may seem more practical to allow the means of the exponential distributions to vary with observed claims. The reason this paper only adjusts the weights is that this allows the correct default distribution to be used when there are no claims. As shown in the previous section, the prior predictive severity distribution will correctly equal the default distribution as long as the expected weights are equal to the desired default weights. However, there is no similar way to do this by varying the means.

For example, suppose desired claim severity is an equally-weighted mixed exponential of means 100 and 300:

$$p(x) = 0.5\frac{e^{-x/100}}{100} + 0.5\frac{e^{-x/300}}{300}.$$

Then $p(x)$ can be expressed as the weighted mixture of various other mixtures of exponentials with means 100 and 300, such as

$$p(x) = \frac{1}{2}\left(0.25\frac{e^{-x/100}}{100} + 0.75\frac{e^{-x/300}}{300}\right) + \frac{1}{2}\left(0.75\frac{e^{-x/100}}{100} + 0.25\frac{e^{-x/300}}{300}\right).$$

However, it is impossible to express $p(x)$ as a (positive) mixture of exponentials with means other than 100 or 300. Thus, the method used in this paper to add uncertainty to a mixed exponential's weights cannot be used to add uncertainty to its means instead.

4. Computation and results

The posterior distribution of the probabilistic model specified in Section 3 is probably not analytically soluble because of the interaction of the Dirichlet and gamma distributions. Interestingly, the model does permit an analytic solution if we omit trend uncertainty, but the number of terms in the analytic solution is exponential in the number of claims and thus is computationally intractable.

However, a Markov chain Monte Carlo (MCMC) technique can be used to quickly compute the posterior distribution. We recommend Just Another Gibbs Sampler (JAGS), which can create an MCMC algorithm given a succinct description of a Bayesian model (Plummer 2003). It is also cross-platform, open source, actively maintained, and easily integrated into R code. Here is JAGS code that expresses the model specified in Equations (1)–(4).

```
model {
  weights ~ ddirch(alpha)
  trend.factor ~ dgamma(trend.shape, trend.rate)

  for (i in 1:n) {
    bucket[i] ~ dcat(weights)
    is.capped[i] ~ dinterval(claim[i], limit[i])
    claim[i] ~ dexp(1 / (means[bucket[i]] / trend.factor^ages[i]))
  }
}
```

Table 3. First 10 samples from MCMC process

Sample #	w1	w2	w3	w4	w5	w6	Trend Factor
1	0.196	0.356	0.214	0.086	0.145	0.002	1.062
2	0.247	0.331	0.218	0.062	0.080	0.062	1.052
3	0.324	0.147	0.160	0.199	0.034	0.136	1.049
4	0.194	0.163	0.421	0.137	0.070	0.015	1.065
5	0.143	0.411	0.169	0.040	0.214	0.023	1.050
6	0.486	0.305	0.118	0.073	0.009	0.009	1.049
7	0.280	0.204	0.206	0.174	0.016	0.120	1.061
8	0.287	0.195	0.435	0.060	0.003	0.020	1.050
9	0.235	0.164	0.389	0.152	0.044	0.016	1.037
10	0.347	0.161	0.189	0.230	0.061	0.011	1.033

Above, `is.capped` is a boolean vector that indicates whether each claim is capped at policy limits. Complete documented `R` code, without the ability to handle adjustable deductible levels, is available in an easy-to-use format from FAViR (2012). Because the example data set was from a casualty line and had no deductible, the above omits deductible handling code for speed. Appendix A contains the full JAGS model.

Appendix A also discusses code to implement the same model written in Stan (Stan Development Team 2013), an alternative to JAGS.

4.1. Example MCMC results

The model was run based on the input data in Section 2, i.e.:

1. The observed 10 claims x_i as shown in Table 1.
2. The default mixed exponential means μ_j and weights a_j as shown in Table 2.
3. The severity uncertainty from Section 3 represented by $\alpha_0 = 20$.
4. The trend mean and standard deviation, chosen in Section 2.4 to be 5% and 1%, respectively.

A few results of the MCMC simulation, samples from the posterior distribution of the bucket weights w_i, are shown in Table 3. Table 4 summarizes all the results.

The required time of the MCMC simulation will increase linearly with the number of claims n, the number of exponential means m, and the number of samples. In this example, 80,000 samples were produced in total. The runtime was quick on a Intel i5 CPU because there were only 10 claims being analyzed. In practice, the runtime of this model does not seem to be a large practical problem—if the number of claims is large enough that the model runs slowly, the number of claims is also large enough that the data has full cred-

Table 4. Prior vs. posterior exponential weights

Prior to Data		Posterior to Data		
Weight (%)	Mean	Weight (%)	Error (%)	Mean
30.0	50,000	30.9	0.04	50,000
25.0	100,000	25.6	0.03	100,000
25.0	500,000	23.3	0.03	500,000
10.0	1,500,000	9.7	0.02	1,500,000
7.0	5,000,000	7.2	0.02	5,000,000
3.0	20,000,000	3.2	0.01	20,000,000
Avg	1,265,000		Avg	1,303,736

ibility and a (faster) maximum likelihood technique, as in Keatinge (1999), would reach the same answer.

Table 4 compares the prior and posterior marginal expected weights. The error column is an estimate of the standard error in the expected value of the MCMC method. This can be decreased through running more simulations. Because the error is estimated using time-series methods, it takes autocorrelation into account. The other exhibits assume this error is acceptably small and can be ignored. The `coda` package of `R` is compatible with JAGS and includes more tools for MCMC error-testing and diagnostics.

Because data from only 10 claims were available, the expected prior and posterior weights are very similar. However, the weights for the $5M and $20M buckets have increased. In general, each claim will nudge up the weights of the high likelihood means. The probability of a sample severity from an exponential distribution being capped at $1M, $e^{-\frac{1Mil}{\mu}}$, increases with the distribution's mean μ. Thus, a claim getting capped at policy limits is evidence in favor of the larger exponential means; conversely, a small claim is evidence in favor of the smaller means. Two of our 10 claims were capped, so our larger means ($5M and $20M) have gotten more posterior weight. The posterior expected trend (not shown in the table) is still 5.0%; this is also not surprising because only a few claims per year were available.

4.2. Sensitivity testing

The results in the previous section depended on the selection of $\alpha_0 = 20$ in Section 3. Because α_0 is proportional to σ^{-2} by Equation (12), a doubling of uncertainty is associated with a fourfold decrease in α_0. Table 5 shows the results of the simulation with α_0 equal to 5 and 80. As α_0 increases, the initial uncertainty decreases and the posterior weights approach the prior weights. The autocorrelation-adjusted error for every reported weight percentage was less than 0.1%.

Table 5. Results by α_0 level

	Weight (%) by α_0 value			
Mean	Prior	80	20	5
50,000	30.0	30.3	31.0	31.7
100,000	25.0	25.2	25.6	27.2
500,000	25.0	24.5	23.3	20.8
1,500,000	10.0	9.9	9.6	9.6
5,000,000	7.0	7.1	7.2	7.5
20,000,000	3.0	3.0	3.2	3.2

5. Applications

The output of this method is a mixed exponential posterior severity distribution and can be used like any other severity distribution. For instance, it can help with pricing excess of loss reinsurance layers, for producing ILFs, or for stochastic risk simulation. The first two applications will be illustrated in this section.

5.1. Reinsurance XOL pricing

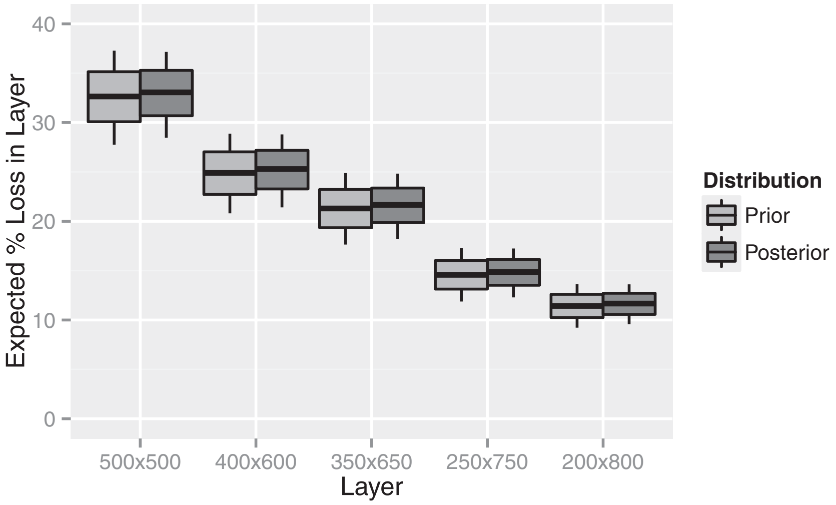
Consider a company that only writes $1M limits and is evaluating the purchase of an excess of loss reinsurance layer. Several options are available: $500K xs $500K, $400K xs $600K, $350K xs $650K, $250K xs $750K, and $200K xs $800K. One way to price these layers that is popular with reinsurers is to compute the percentage loss in each of these layers relative to direct loss.

This ceded loss percentage can be computed by iterating through the sampled weights as in Table 3 and integrating the mixed exponential distribution implied by each set. The results in our example are shown in Figure 1. Each boxplot shows the 10th, 25th, 50th, 75th, and 90th percentiles of the corresponding distribution. Note that these boxplots represent uncertainty about the distribution of claim severities ("parameter risk").

The boxplots do not show the uncertainty caused by whether individual claims fall into the layers ("process risk"), which would depend on the claim frequency distribution. A high anticipated volume of claims would diversify away the process risk, but not the uncertainty shown by the boxplots.

The ceded losses per layer could then be computed by multiplying the calculated percentage loss in each layer by the direct loss ratio and dollars of direct pre-

Figure 1. Prior vs. posterior loss in layer



mium. This ceded loss could then be grossed up for reinsurer expenses and profit load to determine the price of the reinsurance contract.

5.2. ILFs

Another application of a claim severity distribution is calculating increased limits factors. Suppose our company, which currently only writes $1M limits, wants to write policies with a variety of other limits. Just as in the previous section, for each sampled weight distribution we can compute the ratios of expected loss relative to a $1M policy by integrating each specified mixed exponential. The results are shown in Figure 2.

As expected, the uncertainty about loss increases as the limit increases. Because our data set included some claims capped at $1M, the posterior distribution predicts a greater increase in loss for limits above $1M. As before, these boxplots do not include "process risk," only the remaining uncertainty about the underlying claim severity distribution.

Figure 2. Loss relative to $1M limit

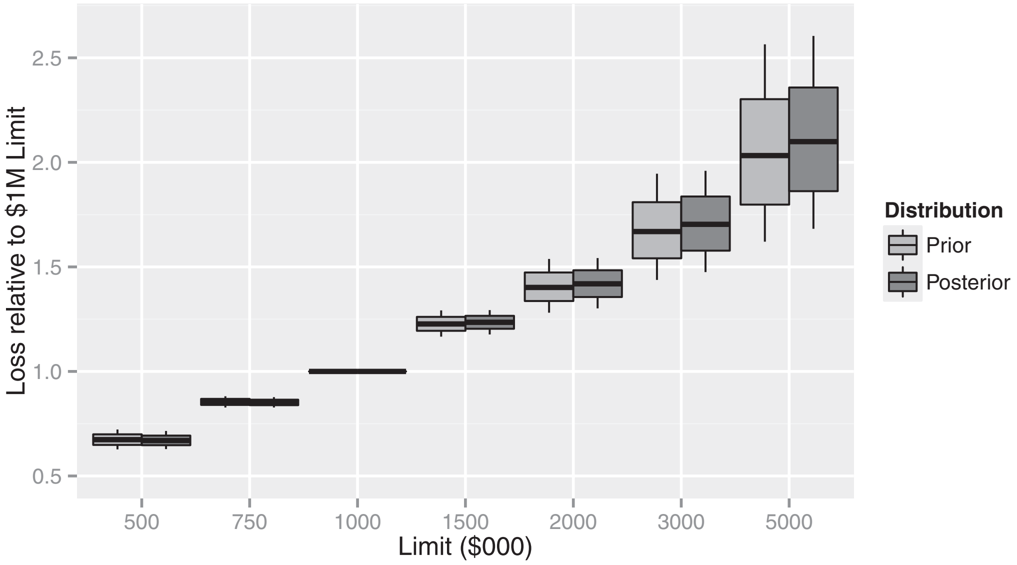


Table 6. ILFs with and without risk load

Limit ($000)	Expected Loss ($000)	Loss Std. Dev. ($000)	Loss-Only ILF	ILF with Risk Load
500	206.1	30.6	0.67	0.63
750	262.9	44.7	0.85	0.83
1,000	309.5	57.6	1.00	1.00
1,500	383.9	81.1	1.24	1.29
2,000	442.9	102.7	1.43	1.53
3,000	535.7	141.8	1.73	1.93
5,000	668.2	207.9	2.16	2.55

The loss relativities in Figure 2 should not be used as ILFs directly because they are not loaded for expense and risk. Another issue not considered here is whether policies with different limits will really be generated by the same severity distribution. Table 6 compares ILFs based purely on expected loss with risk-loaded ILFs based on expected loss plus twice the standard deviation of loss. The additional parameter risk associated with the higher limits leads to more leveraged ILFs.

6. Conclusion

This paper describes a technique for estimating claim severity distributions based on partially credible data and a mixed exponential prior predictive severity distribution. Conceptually the technique is straightforward—we specify a Bayesian model compatible with the prior severity distribution and our uncertainty about that distribution, and then conditionalize on the observed data to find the desired posterior distribution.

The main idea of the Bayesian model is the use of a Dirichlet distribution to parameterize the mixed exponential weights. The Dirichlet distribution only requires one extra parameter, which represents uncertainty about the prior predictive severity distribution. The posterior distribution is computed numerically using MCMC implemented by JAGS.

Once computed, the posterior severity distribution is easy to work with and can be used for such typical applications as pricing XOL reinsurance or computing ILFs.

Acknowledgments

Thanks to Steve Mildenhall, Paul Eaton, and the editors at *Variance* for comments on earlier versions of this paper.

Bibliography

FAViR (Formatted Actuarial Vignettes in R), retrieved January 10, 2012, http://www.favir.net.

Hoff, P. D., *A First Course in Bayesian Statistical Methods,* Springer, 2009.

Keatinge, C. L., "Modeling Losses with the Mixed Exponential Distribution." *Proceedings of the Casualty Actuarial Society* 86, 1999, pp. 654–698.

Klugman, S., H. Panjer, and G. Willmot, *Loss Models: From Data to Decisions,* 3rd ed., Hoboken, NJ: Wiley, 2009.

Lee, S. C. K., and S. X. Lin, "Modeling and Evaluating Insurance Losses via Mixtures of Erlang Distributions," *North American Actuarial Journal* 14:1, 2010, pp. 107–130.

Mildenhall, S. J., "A Multivariate Bayesian Claim Count Development Model With Closed Form Posterior and Predictive Distributions," *Casualty Actuarial Society Forum,* Winter 2006, http://www.casact.org/pubs/forum/06wforum/06w455.pdf.

Plummer, M., "JAGS: A Program for Analysis of Bayesian Graphical Models Using Gibbs Sampling," *Proceedings of the 3rd International Workshop on Distributed Statistical Computing,* Vienna, Austria, 2003, http://www.ci.tuwien.ac.at/Conferences/DSC-2003/Proceedings/Plummer.pdf.

Stan Development Team. "Stan: A C++ Library for Probability and Sampling." (2013) Version 1.1. http://mc-stan.org/.

Wikipedia. "Dirichlet distribution" Retrieved May 2, 2010. http://en.wikipedia.org/wiki/Dirichlet_distribution.

A. Appendix: Additional Model Code

1. Full JAGS Code

The example code used above lacked the ability to handle claims from different deductible levels. The code below handles deductibles, but runs much slower because it must contain an additional inner loop iterating over each exponential distribution.

```
var weights[m], trend.factor, trended.means[n, m],
  ded.weights[n, m], bucket[n];

model {
  weights ~ ddirch(alpha)
  trend.factor ~ dgamma(trend.shape, trend.rate)

  for (i in 1:n) {
    for (j in 1:m) {
      trended.means[i, j] <- means[j] / trend.factor^ages[i]

      ded.weights[i, j] <- weights[j] * exp(-ded[i] / trended.means[i, j])
    }
    bucket[i] ~ dcat(ded.weights[i, ] / sum(ded.weights[i, ]))
    is.capped[i] ~ dinterval(claim[i], limit[i])
    claim[i] ~ dexp(1 / trended.means[i, bucket[i]])
  }
}
```

2. Stan

Although JAGS seemed to work well for this model, another option is Stan (Stan Development Team, 2013), a recent MCMC tool developed by a team led by Andrew Gelman. Unlike JAGS, Stan uses algorithm differentiation to implement the Hamiltonian Monte Carlo technique, which often works better than the traditional Gibbs or Metropolis-Hastings sampling algorithms. Another feature is that Stan model code is turned into C++ and then compiled. Stan model code is more procedural and may be easier to write (for instance, local variables are allowed). Stan, like JAGS, is released under open source licenses.

Unfortunately, currently Stan does not support discrete variables, so the implementation of mixed distributions such as this paper's model is less straightforward. However, the implementation below may be more generalizable to other mixed severity models because it explicitly computes the log likelihood for each observed claim. The exponential distribution used below can be replaced by any distribution with a known density and cumulative distribution function.

```
data {
  // claim data
  int<lower=0> n; // number of observations
  real<lower=0> x[n]; // severity net of deductible
  real<lower=0> deductible[n];
  int<lower=0, upper=1> capped[n]; // 1 if claim is capped at deductible
  real age[n]; // age of claim

  // mixed distribution settings
  int<lower=0> m; // number of mixture distributions
  vector<lower=0>[m] means; // exponential means
  // hyperparameters
  vector[m] alpha; // hyperparameters for dirichlet distribution
  real trend_shape; // shape of gamma trend distribution
  real trend_rate; // rate of gamma trend distribution
}

parameters {
  simplex[m] weights; // exponential weights
  real<lower=.9, upper=1.2> trend; // trend factor (1.0 means no trend)
}

model {
  weights ~ dirichlet(alpha);
  trend ~ gamma(trend_shape, trend_rate);

  for (i in 1:n) {
    vector[m] mu; // trended means
    vector[m] ded_weights; // for weights net of the deductible
    real ps[m]; // for manual addition of mixture likelihoods

    mu <- means / pow(trend, age[i]);

    for (j in 1:m)
      ded_weights[j] <- weights[j] * exp(-deductible[i] / mu[j]);
    ded_weights <- ded_weights / sum(ded_weights);

    for (j in 1:m) {
      if (!capped[i])
        ps[j] <- log(ded_weights[j]) + exponential_log(x[i], 1 / mu[j]);
      else ps[j] <- (log(ded_weights[j])
                    + log1m(exponential_cdf(x[i], 1 / mu[j])));
    }
    lp__ <- lp__ + log_sum_exp(ps);
  }
}
```