Statistical Regularization Applied to Practical Reserving Problems

Gary G Venter and Qiuli Tang

Abstract

More accurate reserve projections can be made with a new statistical approach that has lower estimation and predictive variances than maximum likelihood. Regularization shrinks fitted values towards the overall mean much like credibility does, and also reduces the effective number of parameters used. Simplifying the estimation in this way can help fit more complex models that might represent more underlying features of the data.

We introduce classical and Bayesian versions and a methodology for triangle data, with some code for application packages, and apply it to triangles from the CAS reserve database.

Introduction

Credibility theory shrinks predicted values towards the overall mean in order to improve the estimates. A similar approach is the James-Stein estimator of Stein (1956), who found that if you are estimating more than two values, you always reduce the error variance by such shrinkage. Regularization is a simpler form of shrinkage that first standardizes the independent variables to have mean zero, variance one, and then shrinks the estimated parameters towards zero. The standardization gets reversed by the coefficients and the constant term. The fitted values shrink towards the constant, which is the mean. This is different than using the ratio of within and between variances, like credibility and the James-Stein estimator do, but gets a similar effect.

The two main forms of regularization are lasso and ridge regression. They both use a scaling constant λ to control the degree of shrinkage. Lasso minimizes the negative log-likelihood (NLL) plus λ times the sum of the absolute values of the coefficients, except for the constant, which is not shrunk. Ridge regression does the same thing but with λ times the sum of the coefficients squared. A typical way to set λ is by cross-validation: you split the data into

several subsets, leave these out one at a time, and compute the sum of the NLLs of the left-out points. Some degree of shrinkage is almost always better than straight maximum likelihood estimation (MLE) by this measure.

Of the two approaches, lasso is more popular because some parameter values shrink to exactly zero, so it eliminates variables. There are also Bayesian versions of both approaches. These set the priors for the parameters as mean-zero distributions, which push the parameters towards zero. Modern Bayesian statistical packages do not need to have the posterior distributions specified for this to work – the posteriors are generated numerically.

The Bayesian approach has some advantages:

- MLE comes with penalized likelihood measures like AIC, BIC, etc. that use parameter counts. But shrunk parameters do not count as much, so these measures can over-penalize classical lasso or ridge regression. Bayesian sampling has its own penalized likelihood measure based on cross validation.
- Similarly, MLE has Fisher information for parameter uncertainty, but this is not clear how to apply to shrinkage. Bayesian packages automatically generate parameter distribution samples.
- The Bayesian approach can also set a prior for λ, which both estimates λ and samples from a range of λ values instead of just a single λ, which can improve the estimation.
- The frequentists versions, as we will see below, actually calculate the Bayesian posterior mode in a deterministic manner. But Bayesians tend to feel that the posterior mean is more likely to represent the underlying population the sample is drawn from.

We will outline both approaches, but the examples will use the Bayesian approach.

Triangle models need to have a parameter for each row, column, and perhaps diagonal, but these can be put on smooth curves to use fewer parameters. We follow Barnett and Zehnwirth (2000), who put the row, column, and diagonal parameters on separate piecewise-linear curves, and then estimate the slope changes between the line segments. The slope changes produce the row and column factors, and parameter shrinkage on them smooths the piecewise-linear curves. If a slope change shrinks to zero, that just continues the previous line segment through that point. Venter and Şahin (2018) apply this with Bayesian shrinkage to fitting mortality data, which is just a big triangle. We apply much of that methodology to reserving models from an applied perspective. Similarly, Gao and Meng (2018) apply Bayesian shrinkage to a cubic-spline fit of a loss reserve model. That can give smoother curves but seems more complex to implement.

Parameter Shrinkage – Bayesian and Classical Formulas

The results of Bayesian and classical regularization are often similar, but the approaches start from a philosophical difference: in classical statistics, parameters are constants, but for Bayesians they have distributions. The two are converging, however, due to a classical method called random effects.

That method postulates statistical effects across a population, like difference of territory frequency from average, that are assumed to average to zero. With mean zero, their distribution might be described by a single dispersion parameter, like the normal variance. The random effects themselves are not parameters, and they are not estimated. However the individual values can be projected from the data. Usually most of them are believed to be at or near zero, so it takes a degree of evidence in the data to project one of them to be positive or negative. This is almost a vocabulary issue – say you are projecting random effects instead of estimating random parameters, and you have a distribution of effects instead of a prior distribution for the parameters.

To illustrate the math of it, assume that the effects are double exponential, or Laplace, distributed. This is a distribution that looks like an exponential for positive values, and its mirror image over the y-axis for negative values. The density for a variable β is

$$f(\beta|\lambda) = 0.5\lambda e^{-\lambda|\beta|}$$

This has variance $= 2/\lambda^2$ and kurtosis $= 6$. Say there are k random effects β_j, plus perhaps other parameters, including λ.One way to simultaneously estimate the parameters and project the effects is to maximize what they call the joint likelihood, which is the likelihood of the data, given the parameters and the random effects, times $\Pi f(\beta_j|\lambda)$.

The negative of the log of $f(\beta_j|\lambda)$ is just $log(2) - log(\lambda) + \lambda|\beta_j|$. Then maximizing the joint likelihood becomes minimizing

$$NLL + \lambda \sum |\beta_j| - log(\lambda)k$$

If you fix a value of λ, the last term does not affect the minimization, and dropping it projects the random effects for that λ by the lasso formula. If that term is kept in, the value of λ produced by the minimization is also an estimate of that parameter.

This connects with the Bayesian approach. For data X and parameters β, Bayes Theorem is:

$$p(\beta|X) = \frac{p(X|\beta)p(\beta)}{p(X)}$$

The left side is the posterior distribution of the parameters given the data, and the numerator of the right side is the likelihood times the prior. Here the β are parameters, but this numerator is the same mathematical formula as the joint likelihood in the random effects case. The denominator $p(X)$ is a constant for a given dataset, so maximizing the numerator maximizes the posterior. Thus the random effects solution gives the posterior mode, and for the Laplace prior gives classical lasso. This is why the use of the Laplace prior is called Bayesian lasso. Doing all this normal distribution, instead of the Laplace, for the random effects gives ridge regression.

One detail here is that in the Bayesian case this works for a fixed value of λ. If λ is to be estimated as well, it also must be given a prior. If it has a uniform prior $= K$ over some interval, then $log(K)$ is subtracted from the log of the prior. But since that is a constant, it does not get into the minimization, and so the posterior mode is still at the minimum of $NLL + \lambda \sum |\beta_j| - log(\lambda)k$. Note that as λ increases, the parameters are pushed more towards zero to compensate, but that makes the NLL get higher. At the same time, $-log(\lambda)k$ is decreasing. Thus at some point they all balance at a minimum.

Bayesian Estimation

This section outlines how the numerical Bayesian approximation to the posterior distribution of the parameters is calculated, software implementation of that, and goodness of fit measurement.

The numerical generation of samples of the posterior distribution only requires the priors and the likelihood. The posterior distributions come out from the process, so do not need to be specified in advance. The main methodology for this is called Markov Chain Monte Carlo, or MCMC. That just means that samples are generated stochastically, and each sample is generated from the previous one, without reference to earlier samples. Only the numerator of Bayes formula is needed, as the denominator is a constant.

The original MCMC methodology was the Metropolis sampler. It starts with a proposal generator to create a possible sample of the parameters from the latest accepted sample. If this produces a higher value for the posterior probability, the sample is added to the collection.

If it doesn't, there is an acceptance rule to put the sample in or not, based on a random draw. After a warmup period, the retained samples are representative of the posterior.

A refined version, the Hastings-Metropolis sampler, is more efficient. Further refinements include Hamiltonian mechanics and the no-U-turn sampler, which evolve the proposal generator dynamically. They are used in the Stan MCMC package, which has R and Python versions. Another MCMC algorithm is the Gibbs sampler, which draws parameters sequentially from the posterior distribution of each parameter conditional on the data and the latest sample of all the other parameters. The JAGS (just another Gibbs sampler) package uses it. Basically then, MCMC is looking for parameters that give relatively high values to the loglikelihood plus the sum of the logs of the prior probabilities of the parameters.

Traditional goodness-of-fit measures, like AIC, BIC, etc., penalize the loglikelihood with parameter-count penalties, but parameter shrinkage distorts the parameter count. For lasso, the gold standard of model testing is leave-one-out estimation, or loo. The model is fit over and over, each time leaving out a single observation, with the loglikelihood computed for each omitted point. The sum of those loglikelihoods is the loo fit measure.

But loo is computationally expensive. To address this, Gelfand (1996) developed an approximation for a point's out-of-sample likelihood from an MCMC sample set using the numerical integration technique importance sampling. In his implementation, a left out point's likelihood is estimated as its weighted average likelihood across all the samples, taking the weight for a sample to be proportional to the reciprocal of the point's likelihood under that sample. That gives greater weight to the samples that fit that point poorly, which would be more likely to occur if the point had been omitted. The estimate of the probability of the point comes out to be the reciprocal of the average over all the samples of the reciprocal of the point's probability in each sample, which is actually the harmonic mean of the point's likelihood across the samples. With this, an MCMC sample of the posterior distribution is enough to estimate loo.

This gives good but still volatile estimates of the loo loglikelihood. Vehtari, Gelman, and Gabry (2017) address that by something like extreme value theory – fitting a Pareto to the probability reciprocals and using the fitted Pareto values instead of the actuals for the largest 20% of the reciprocals. They call this "Pareto-smoothed importance sampling." It has been extensively tested and is becoming widely adopted. Their penalized likelihood measure is labeled $\widehat{elpd}_{loo}$, standing for "expected log pointwise predictive density." It aims at doing what AIC etc. are aiming to do also – adjust the loglikelihood for sample bias.

The Stan software provides a loo estimation package that can work on any posterior sample,

even those not from Stan. It outputs $\widehat{elpd}_{loo}$ as well as the implied loglikelihood penalty and something they call looic – the loo information criterion – which is $-2\widehat{elpd}_{loo}$ in accord to standards of information theory. Since the factor of 2 is not critical, here the term looic is used for $-\widehat{elpd}_{loo}$, which is half of the usual looic but conveniently is the NLL increased by the parameter penalty.

The derivation of looic, like that of AIC, starts by assuming that the data is generated by the model. This is problematic in financial applications, where the data comes from complex economic processes that the model is just trying to impose a degree of structure onto. One response is to use slightly more parsimonious models than the statistical measures suggest, and that could be advisable here.

An increasingly popular shrinkage prior is the Cauchy distribution, with $1/p(b) = \pi(\lambda^2+b^2)/\lambda$ and $-log(p(b)) = -log\lambda + log\pi + log(\lambda^2 + b^2)$. For a fixed λ, the posterior mode minimizes $NLL + \sum log(\lambda^2 + b_j^2)$. This is an alternative to both lasso and ridge regression. The Cauchy prior often yields more parsimonious models than the normal or Laplace priors do. It can have a bit better or bit worse penalized likelihood, but even if slightly worse, the greater parsimony makes it worth considering. It has more weight near zero but is also heavier tailed, which pushes parameter more towards zero, but allows a few larger parameters when they are called for. It also seems to produce tighter ranges of parameters.

Bayesians often prefer the posterior mean to the posterior mode. They seem to have an inherent suspicion that the parameters that maximize the posterior probability could be doing so by over-fitting the particular sample. The posterior mean averages all parameter sets that provide a plausible explanation of the data. The posterior mean does not optimize a goodness-of-fit measure for the data – in fact any such measure runs the risk of sample bias. It does minimize the squared parameter error. The posterior mode optimizes the lottery number measure of parameter error: all misses from the exactly right answer are equally bad. But for parameters, close is usually better.

Loss Reserving – Models and Estimation

The basic loss reserving model estimates mean losses for a cell as the product of row, column, and possibly diagonal factors. Often this is estimated in log form, so the log of the cell mean is the sum of row, column, and diagonal impacts. Row-column-diagonal models do not always specifically include a constant term, as that can be taken as the first parameter in one of the directions, but here we make it a parameter, since regularization models include a constant

term that is not shrunk. The model for the fitted value for the cell in row w, column u, in log form, is then

$$\mu_{w,u} = c + p_w + q_u + r_{w+u}$$

where p_w, q_u, r_{w+u} are the row w, column u, and diagonal $w+u$ parameters, respectively, with lags u starting at zero. Often $\mu_{w,u}$ represents the log of the cell mean, but it could be a parameter of the fitted distribution, like a lognormal or gamma parameter. Fitting piecewise-linear curves to this model was introduced to the actuarial literature in Barnett and Zehnwirth (2000), although actuaries were doing it earlier. In social sciences, the row-column-diagonal model goes back to Greenberg, Wright, and Sheps (1950), where it is called the age-period-cohort model.

After exponentiating, the linear parameters become factors that are multiplied together. In a recent Variance article, Müller (2016) suggests expanding the multiplicative model with an additive component. He argues that some part of loss development is from late reported claims, and these could be more related to exposure than to losses emerged. Any accident-year exposure variable, such as premium or policy count, could be used as the exposure base. That would be multiplied by column coefficients, then added to the multiplicative mean for the cell. A constant exposure for all the rows could be used if more detail is not available, or if the data already has been divided by an exposure variable, like accident-year premiums. Also the coefficients could be on a curve fit across the columns. This could be done in log form for fitting. After exponentiating, the model for $\mu_{w,u}$ would be:

$$\mu_{w,u} = A_w B_u C_{w+u} + D_u E_w$$

where E_w is the exposure for accident year w (or just a constant) and the D_u are column parameters. Here the first term is the exponentiation of the linear model. We try this model in the examples, with D on a piecewise-linear curve. It can actually capture a broader range of effects than just IBNR claims. An additive constant by column, for instance, can make each column unbiased, which it might not be even if the log values are. Also the interaction of two sets of column parameters can pick up changes in payout patterns over the accident years, as discussed in G. Meyers (2015).

Distributions for Residuals by Cell

For residual distributions in the loss reserving examples, we use the gamma distribution. If the parameters for cell j are a_j and b_j, the mean and variance are $a_j b_j$ and $a_j b_j^2$, and the

skewness is $2/\sqrt{a}_j$. The variance then equals both $b_j \text{mean}_j$ and mean_j^2/a_j. If a_j is fixed across the cells, then the variance is proportional to the mean squared, but if b_j is fixed, the variance is proportional to the mean.

The latter generally appears to be more common in loss triangles, and is the assumption underlying the over-dispersed Poisson (ODP) model. The ODP distribution in the exponential family is defined only on integer multiples of the Poisson mean, so is not usually appropriate for loss reserve data. Still, the ODP can be used with GLM software under the assumption that the distribution is not the ODP of the exponential family, but has the same quasi-likelihood, which is what GLM uses in the fits.

Since the assumed distribution is no longer in the exponential family, this fitting is not maximum likelihood. Also, without the density function, Fisher information is not available for parameter uncertainty, so bootstrapping, often with questionable assumptions, is used. But there is no ODP distribution available for simulation, so the ODP is usually approximated by a gamma with the same mean and variance. The gamma skewness is twice that of the ODP with that mean and variance. That actually turns out to be more reasonable in many cases, however, as the ODP is often not skewed enough.

Since in the end the gamma is being used by cell anyway, it is more direct just to fit the gamma, assuming constant b. This cannot be done with variance proportional to mean with GLM software, but it can be done with either MLE or MCMC. With MLE estimation, Fisher information can be used for parameter uncertainty. Under MCMC, parameter ranges come out automatically. This is used in the examples.

For the degree of shrinkage, we selected a uniform prior for the log of the Laplace scale parameter $s = 1/\lambda$ on the range $[-5, -0.2]$. This allows a maximum of about 0.8 for s. Letting s get too high can create convergence difficulties. Usually it ended up quite a bit less than 0.8, so the prior was not really a constraint.

The constant term and the gamma b parameter were not shrunk. For the constant, we settled on a uniform prior on $[-4, 16]$. We started with a wider prior but it allowed some very poor local maximums. The constants for the datasets studied ended up well inside this range. We used a uniform prior on $[-20, 20]$ for the log of the gamma b. When giving a positive parameter a wide range, there is a risk that the parameter will be towards the lower end of the range, and the wide range could push it to be too high. Putting a uniform prior on the logs usually prevents this.

The estimation can be set up in regression form by stringing out the data array into a vector, with the explanatory variables put in a design matrix with entries for each cell. It is

helpful to record three columns that identify the accident year, lag, and calendar year for each observation. If estimating the level parameters by regression, each parameter can be represented as a dummy variable with value 1 for a cell that it affects, and 0 for other cells. For the second difference variables, the dummies are more complicated.

The second difference parameters, starting from the first accident year, lag, and calendar year, add up cumulatively to the first differences of the level parameters, which in turn add up to the level parameters. The dummy variable value for a second-difference parameter for a given cell is the number of times that the second difference gets added up for the cell. If a_u is the second difference parameter for lag u, and a cell is at lag i, the a_u dummy variable has value $\max(0, 1 + i - u)$ at that cell. This is the same for the accident and calendar year second difference dummies. Putting all those together defines the design matrix.

Examples

G. G. Meyers and Shi (2011) have compiled annual statement loss reserve triangle data for several lines of insurance and many carriers for accident years 1988-97. We look at commercial auto incremental paid loss triangles for Florida Farm Bureau, State Farm and USAA, after dividing each row by its accident-year premium. Since the development is largely complete after nine years, we use 10×9 triangles. Dividing by premiums makes them loss-ratio triangles, and eliminates the need to use accident-year parameters for the known effects of premium changes. Incremental triangles are used because cumulative triangle columns are almost always highly correlated, violating regression assumptions. Incremental triangles could be negatively correlated, but this turns out to be rare in practice.

State Farm

The State Farm triangle is shown in Table 1. From this, all the years appear to have similar loss ratios. The highest year varies by column. The second column has lower payments than the first, and these seem to get even lower for the more recent accident years – which could be from a change in the payment pattern. The later columns drop off fairly quickly.

We model this first by row and column, then column and diagonal, then by all three. A loss ratio triangle may show stronger diagonal effects than row effects, but this was not the case here. We fit the row-column model by first including all of the row and column slope-change variables, and then eliminating any that had coefficients near zero with a high

Table 1: State Farm Commercial Auto Incremental Paid Loss Ratio Triangle

	1	2	3	4	5	6	7	8	9
1988	0.1910	0.1873	0.1242	0.0730	0.0514	0.0204	0.0092	0.0061	0.0027
1989	0.1945	0.1919	0.1029	0.0761	0.0350	0.0218	0.0128	0.0060	0.0064
1990	0.2017	0.1989	0.1274	0.0784	0.0375	0.0277	0.0097	0.0051	
1991	0.1862	0.1779	0.1136	0.0716	0.0378	0.0199	0.0092		
1992	0.1902	0.1858	0.1050	0.0675	0.0360	0.0239			
1993	0.2032	0.2031	0.1148	0.0645	0.0308				
1994	0.2159	0.1822	0.1012	0.0601					
1995	0.2087	0.1737	0.0966						
1996	0.1960	0.1571							
1997	0.1865								

standard deviation. If that hurt the penalized likelihood looic measure, we put them back in. This ended up retaining all but the ninth row variable and all but the fourth column. Looic was 189.6, with a parameter penalty of 10.7, and so NLL of 178.9. With a constant term, a gamma b variable, but no parameters for the first row or column, this left nominally 17 variables, 15 of which were subject to shrinkage. Usually the parameter penalty is higher than the effective number of parameters used, considering shrinkage. Perhaps there were the equivalent of seven or eight non-shrunk parameters here. The resulting row and column factors are shown in Figure 1. (The fitted values are the product of the constant with these row and column factors.)

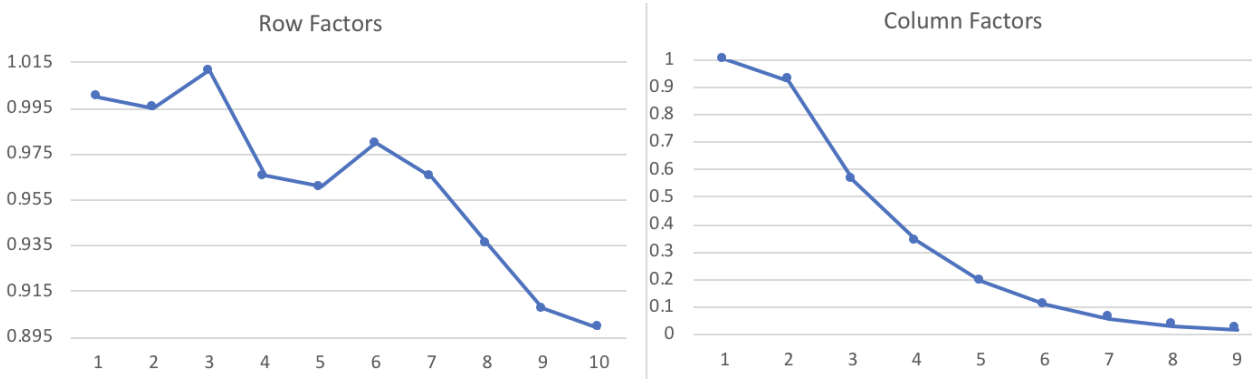

Figure 1: State Farm Row-Column Model Implied Factors

The row factors are in a small range, which is consistent with a nearly constant loss ratio. The column factors are on a fairly smooth curve, which is a typical result from shrinking the slope changes.

The Stan code we used is shown below. Most of it consists of setup and variable definitions.

```
data {
  int N;           //number of observations
  int U;           //number of variables
  vector[N] y;     //the triangle in a column
  matrix[N,U] x1;  //design matrix with U columns
}
parameters {  // all except v will get uniform prior, which is default
 real<lower=-4, upper=16> cn;   //constant term, in wide enough range
  vector[U] v;                  //the parameters
 real<lower=-5, upper = -0.2> logs;     //log of s, related to lambda
 real<lower=-20, upper = 20> logbeta;   //log of gamma b parameter
}
transformed parameters {
 real beta;                        //gamma b parameter
 real s;      //Laplace variance parameter, like 1/lambda
 vector[N] alpha;                  //fitted gamma a parameters by cell
 beta = exp(logbeta); //for positive parameter, uniform on log is better
 s = exp(logs);   //s tends to end up on an exponential scale
 alpha = exp(x1*v+cn)*beta; //vector of gamma a parameters = mean*beta
}
model {  // gives priors for those not assumed uniform. This one for lasso.
   for (i in 1:U)  v[i] ~ double_exponential(0, s);
for (j in 1:N) y[j] ~ gamma(alpha[j], beta);  //Stan gamma mean is a/b
}
generated quantities { //outputs log likelihood for looic
  vector[N] log_lik;
for (j in 1:N) log_lik[j] = gamma_lpdf(y[j] | alpha[j],beta);
}
```

The data section reads in the variables that have already been defined in the R workspace. The model itself starts on the last line of the transformed parameters section. It computes the gamma a for each cell by multiplying the slope-change variables by the parameters and adding the constant, then exponentiating to get the cell mean and multiplying by b to give a. In Stan, the gamma mean is a/b. The model section describes the Laplace prior and the gamma likelihood. The generated quantities section is needed to provide the loglikelihoods used in calculating looic. The name of this file is "logregressiongam.stan".

Next is the R code we used to set up Stan, run it, and look at the output:

```
setwd("/Users/yada/yada/yada")
library(rstan)
rstan_options(auto_write = TRUE)
options(mc.cores = parallel::detectCores())
library("loo")

y = scan('sfarm_y.txt')   #scan reads a txt file into a vector
library(readxl)           #helps in reading Excel files
x1 = as.matrix(read_excel("sfarm_ac.xlsx"))
U = ncol(x1)
N = length(y)
c(N,U)

fit2 = stan(file = 'logregressiongam.stan', verbose = FALSE,
            chains = 4, iter = 2000)
print(fit2, pars=c("cn", "v", "beta", "s"),
      probs=c(.05, 0.2, 0.5, 0.8, 0.95), digits_summary = 3)
plot(fit2, pars =("v")

out <- get_posterior_mean(fit2)
write.csv(out, file="out_ac_sfarm.csv")

log_LL <- extract_log_lik(fit2)
loo_LL <- loo(log_LL)
loo_LL
```

The triangle is in a single column in a txt file without a header. The Excel file with the dummy variables has column headings. We ran Stan with 2000 iterations in four parallel chains. Stan checks convergence by comparing the chains. The default is to use half of the iterations for warmup and the second half for sampling. The remaining lines of code give some useful output.

We next tried a column-diagonal regression, but this did not fit as well. The same exact code works for that, just with dummy variables by column and diagonal instead of by row and column. We also tried adding diagonal parameters to the row-column model, but this did not improve the penalized likelihood. Then we tried an additive component, as in Müller

(2016), and this provided a significantly improved fit.

We just used a constant of unity for the exposure level for each row. Since the rows were divided by accident-year premium, that makes premium the exposure measure. The additive part makes a portion of each column a constant that does not depend on the row factors. In this case the penalized loglikelihood dropped to 124.1, with a parameter penalty of 9.1 and NLL of 115. There was a slope-change parameter for every column's exposure factor (including now for the first column), so the drop in the parameter penalty from 10.7 is interesting. The penalty is less if the left-out observations are better predicted. Smoother curves also have fewer effective slope-change parameters, which can also reduce the penalty.

The implied factors and constant terms for this and the simple factor model are shown in Figure 2. The column constants are the accumulation of the slope changes but are not exponentiated, and are not even forced to be positive. They are simply added to the row × column products.

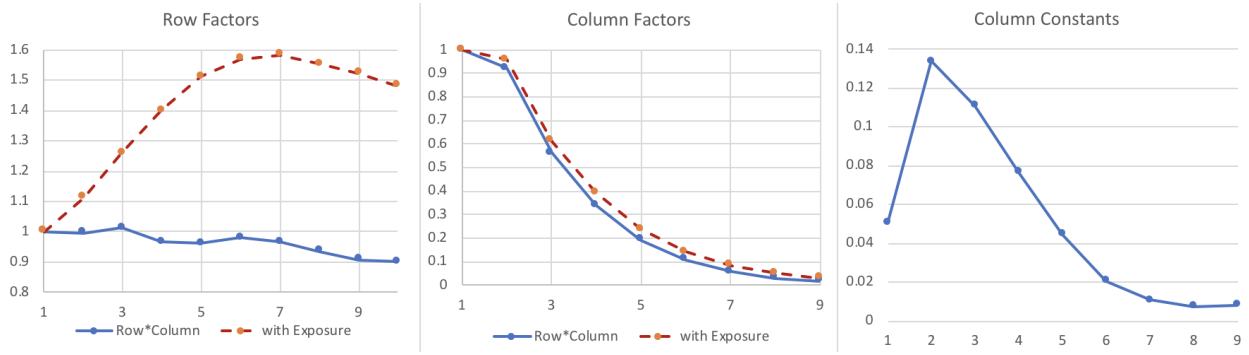

Figure 2: State Farm Row-Column Model with Exposure Term

In the exposure model, the row factors increase for the first seven accident years. The additive part of the columns is not affected by that, so the fitted values do not increase as fast, and this can vary by column. A shift in the payout pattern can be picked up by this model, and that could be what is giving the much better fit.

USAA

The USAA triangle is shown in Table 2. It is unusual for a loss ratio triangle in that the losses get a lot smaller in the later accident years, especially in the later columns. Perhaps they had strong rate changes during these years, a lessening of the inflation impact, or a change in business mix. The losses also seem to pay out more slowly than for State Farm. This could

Table 2: USAA Commercial Auto Incremental Paid Loss Ratio Triangle

	1	2	3	4	5	6	7	8	9
1988	0.1048	0.3011	0.1936	0.1366	0.0809	0.0332	0.0186	0.0027	0.0040
1989	0.0994	0.2916	0.2413	0.1574	0.0994	0.0452	0.0077	0.0090	0.0026
1990	0.0359	0.3134	0.1998	0.1304	0.0837	0.0215	0.0108	0.0024	
1991	0.0308	0.1963	0.1941	0.1084	0.0502	0.0171	0.0057		
1992	0.0538	0.2138	0.1815	0.0985	0.0354	0.0200			
1993	0.0261	0.2524	0.1043	0.0535	0.0302				
1994	0.1728	0.1933	0.0805	0.0542					
1995	0.0835	0.2799	0.1242						
1996	0.1084	0.1970							
1997	0.0920								

be from a different mix by state. In any case, we followed the same fitting sequence as for State Farm.

The row-column model had looic and NLL of 120.7 and 111, which isn't as good as the column-diagonal model, which came in at 112.8 and 101.2. It is not unusual in loss ratio triangles to get a better fit from the diagonal factors. Again the model with all three dimensions did not improve on this. The column-diagonal model used all the diagonal parameters and all the columns except 7 and 9.

We also tried a Cauchy prior for this model, and it came in with a higher looic of 116.9 but used fewer parameters – diagonals 6, 7, 9, and 10 were not used, nor were columns 7, 8, and 9. The loo parameter penalty was 7.5, compared to 11.6 for the Laplace prior. This is the kind of slightly-worse-fitting but more parsimonious result that should be considered as an alternative if the processes that generate the data are considered to be subject to change. The factors from the two priors are graphed in Figure 3. The diagonal factors in particular show the more parsimonious nature of the Cauchy fit.

The lower loss ratios for the more recent accident years are explained here as a downward trend across the calendar years. The loss ratios do appear to drop more in the later columns, which is what a calendar-year effect would produce. The downward trend suggests that rate changes were greater than inflation, which could have been caused by a unexpected decline in inflation, for example.

We also fit an exposure trend to this data. This produced a bit better fit, with looic of 110.1, but NLL was worse at 103.5. Looic, the penalized loglikelihood, is the better measure. The parameter penalty was only 6.6 for the exposure model, so it provided a better fit on omitted data points. But the parameters were surprising, in that all the column factors dropped out.

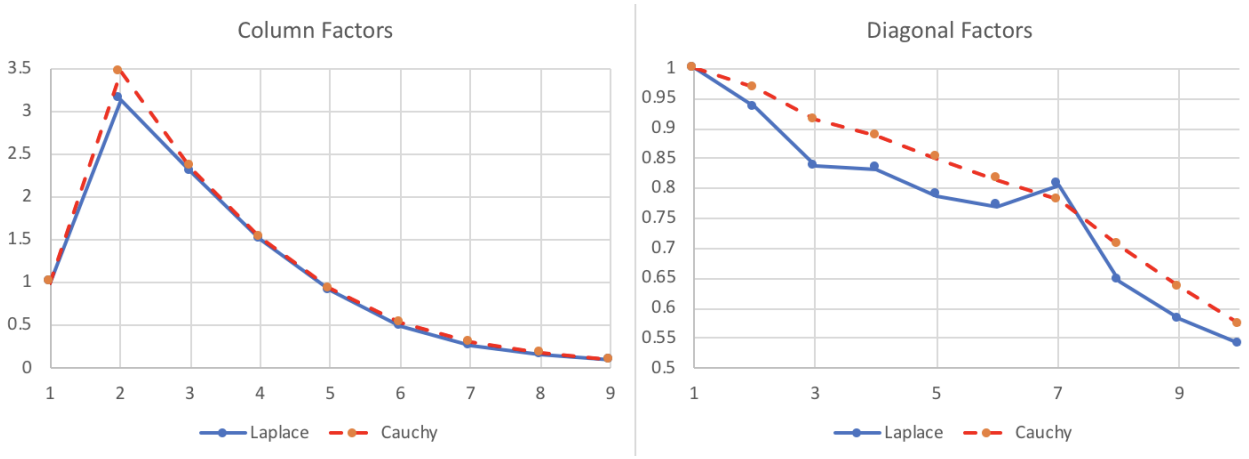

Figure 3: USAA Column-Diagonal Model

Table 3: Florida Farm Bureau Commercial Auto Incremental Paid Loss Ratio Triangle

	1	2	3	4	5	6	7	8	9
1988	0.3351	0.1471	0.0987	0.0496	0.0264	0.0181	0.0083	0.0010	0.0003
1989	0.2109	0.2228	0.1419	0.0555	0.0255	0.0263	0.0050	0.0013	0.0026
1990	0.2201	0.2013	0.1117	0.0553	0.0222	0.0043	0.0117	0.0010	
1991	0.1974	0.1554	0.1601	0.0474	0.0123	0.0079	0.0004		
1992	0.1416	0.2223	0.0949	0.0517	0.0496	0.0142			
1993	0.2340	0.2053	0.0659	0.0357	0.0220				
1994	0.2658	0.1875	0.1020	0.0686					
1995	0.2451	0.2427	0.0710						
1996	0.2242	0.1704							
1997	0.1949								

The result is a purely additive model – diagonal factors $\times$ the constant plus the column constants. Additive models have been suggested occasionally, but it generally has not seemed worth the effort to try them routinely. Here they can pop up as the result of the exposure model. The Cauchy prior in this case produced basically the same fit as the Laplace.

Florida Farm Bureau

Table 3 shows the triangle for the Florida Farm Bureau. Payments are highest at the first two lags then fall off. There is perhaps a shift towards earlier payments over time. For instance, the third lag averages 13% for the first four accident years, and 8% for the next four. The first-to-second incremental development factor averages 95% for the first five accident years and 83% for the next four, although the factors are fairly volatile.

Table 4: Florida Farm Bureau Fit Comparison

	Looic	NLL	Penalty	Parameters
Row-Column	142.0	132.0	10.0	16
With Exposures	134.1	123.3	10.8	20

The row-column model gave a better fit than did including diagonal factors, and again including an additive exposure component significantly improved the fit. Table 4 is a comparison of the fits. Most of the slope-change parameters are near zero, so have probably been shrunk in the fitting. The parameter penalties are smaller than the nominal parameter counts but would be larger without shrinkage. There are nominal parameters for most rows and columns, with four more when exposure constants by column are included. The penalty isn't much higher with exposures, probably because they improve the out-of-sample fits, possibly with more shrinkage.

The factors and constants are graphed in Figure 4. The row factors increase over the accident years in the model with exposures, as they did for State Farm. This makes the earlier rows lower, especially in the first two columns where the column factors are high. This looks like a shift in the payout pattern in comparison to the row-column model.

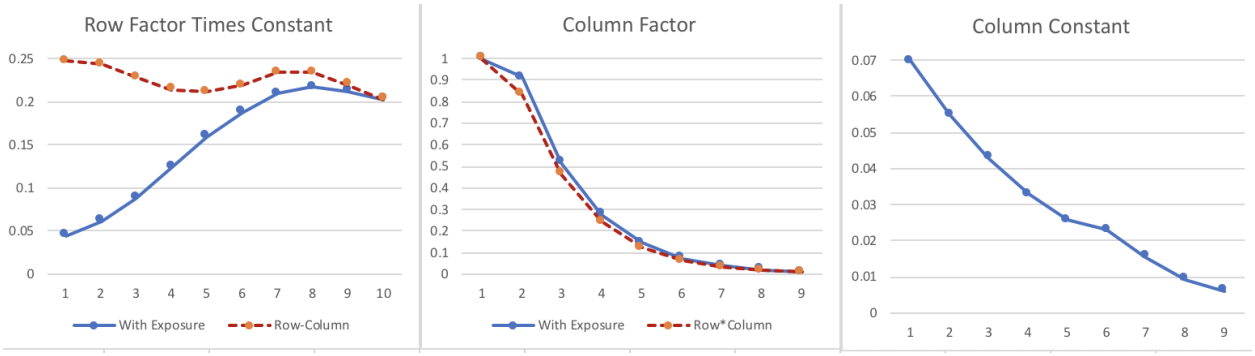

Figure 4: Florida Farm Bureau Model Factors

Modeling Multiple Matrices

Another application of shrinkage is simultaneous modeling of related data sets. We illustrate one way to do this, using the State Farm and Florida Farm triangles.

Assuming the rows and columns of both triangles are strung out in the same order into a vector, start by making the dependent variable a column with the two original columns put one on top of the other. Here we put Florida Farm on top. The model is structured as a

common set of parameters that apply to both companies, and then a set of adjustments that only apply to a single company, in this case to State Farm.

For the design matrix for the common parameters, use the one for the first company that would be used for it by itself in the top half, and duplicate it in the bottom half. Use this again in another group of columns for the bottom half again, and make those columns zero for the top half. Add a column of zeros for the top half and ones for the bottom half to represent an adjustment constant for State Farm that is subject to shrinkage. The second group of columns is for the adjustments for the second company. Hopefully these factors will come out smaller, so with more shrinkage and so fewer effective parameters. This is a sort of credibility weighting, where the degree to which the second company deviates from the first is shrunk by the shrinkage priors.

We based this example on the row-column fits, which were good for these two companies. The variables used were all the rows and columns that were in the final model for either company. A few of these variables were eliminated in the fitting, where we again took out variables with parameters near zero that had wide ranges of fitted values, especially if so doing did not increase looic. For the common parameters, all the rows were retained except for 7 and 10. Columns 7, 8, and 9 were eliminated. The State Farm adjustments kept the constant term, rows 3, 4, 6, 7, 9, 10 and columns 2 – 4. The combined model had a looic of 308.0 and a penalty of 15.7, compared to totals of 331.6 and 20.7 for the individual models, so is better on both. A lower penalty giving a lower looic was the goal of the joint fitting. The better NLL is more of a surprise, given that it is a more parsimonious model.

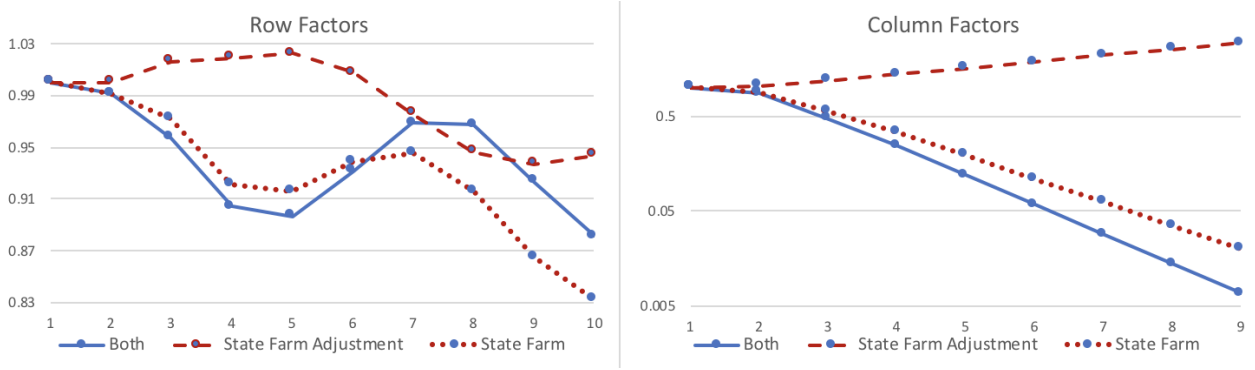

Figure 5: Joint Model Factors

The parameters are slope changes in the log scale, and for State Farm, the adjustments add to the combined parameters. The combined parameters are used for Florida Farm as is. After adding up to the row and column level parameters and exponentiating, the State Farm adjustments become factors applied to the common factors. The resulting factors are

all shown in Figure 5, with the columns on a log scale. The slope changes were indeed closer to zero for the adjustment than for the original factors in either model.

We tried this same method on the model that includes an additive exposure term. It came out better than the joint model here, but not as good as the individual models combined. The additive term produced a fairly large improvement for State Farm, and the joint model could not match that. You would think that it could do so just by making all the adjustments get back to the separate models, but apparently that does not give the posterior mean with all these shrinkage priors. The differences between the companies now have shrinkage priors, which prioritizes models that are similar for the two companies.

The posterior probability is the prior times the likelihood, divided by a constant that does not enter into the calculations. When the shrinkage priors apply to the differences between the companies, models where these differences are larger get lower posteriors, even if they have higher likelihoods.

This could also help explain how the joint model could have a higher likelihood than the individual ones combined. If the individual models each need a similar parameter that is fairly far from zero, like the change from the second to third column, that would decrease the posterior probability for both models. In the joint model this would only happen one time, so would not be as much of a penalty. Thus the joint model could get to parameters that have a higher likelihood for both companies that would be penalized more in the individual models by the lower prior probabilities. Thus it is worth trying both joint and individual models.

Conclusion

Regularization is a model estimation approach used to reduce predictive variance. The Bayesian form improves the selection of the degree of shrinkage, and provides model fit comparisons and range estimates. One way to use it on reserving models is by shrinking or eliminating second-differences of piecewise-linear curves through the usual parameters. We applied this to row, column and diagonal models in the examples, using gamma-distributed residuals with the variance proportional to the mean, and with Laplace and Cauchy priors.

This parameter reduction also facilitates using more complex models, and we showed how to apply it to the model with an additive exposure component recently introduced in Variance by Müller (2016). This model provided the best fit for all the example companies. It even can produce an adjustment for shifts in the payout pattern.

Finally, we used this methodology for joint modeling of multiple triangles, which could for example be applied to sublines of a larger line as a form of credibility weighting them together. Although we only did this for two triangles, there are various ways it could be generalized to more, either by having all but one triangle have adjustments to the common factors or by grouping some triangles so that all but one of those has factors that differ from the rest of the subgroup. This is a promising area for further research, especially if a number of related triangles are available.

References

Barnett, Glen, and Ben Zehnwirth. 2000. "Best Estimates for Reserves." *Proceedings of the Casualty Actuarial Society* 87: 245–303.

Gao, Guangyuan, and S. Meng. 2018. "Stochastic Claims Reserving via a Bayesian Spline Model with Random Loss Ratio Effects." *Astin Bulletin* 48:1: 55–88.

Gelfand, A. E. 1996. "Model Determination Using Sampling-Based Methods." *Markov Chain Monte Carlo in Practice, Ed. W. R. Gilks, S. Richardson, D. J. Spiegelhalter* London: Chapman and Hall: 145–62.

Greenberg, B. G., John J. Wright, and Cecil G. Sheps. 1950. "A Technique for Analyzing Some Factors Affecting the Incidence of Syphilis." *Journal of the American Statistical Association* 45:251: 373–99.

Meyers, Glenn. 2015. "Stochastic Loss Reserving Using Bayesian Mcmc Models." *CAS Monograph Series* 1: i–55.

Meyers, Glenn G., and Peng Shi. 2011. "CAS Loss Reserve Database." *Available at CAS Website* http://www.casact.org/research/index.cfm?fa=loss_reserves_data.

Müller, Thomas. 2016. "Projection for Claims Triangles by Affine Age-to-Age Development." *Variance* 10:1: 121–44.

Stein, Charles. 1956. "Inadmissibility of the Usual Estimator of the Mean of a Multivariate Normal Distribution." *Proceedings of the Third Berkeley Symposium* 1: 197–206.

Vehtari, Aki, Andrew Gelman, and Jonah Gabry. 2017. "Practical Bayesian Model Evaluation Using Leave-One-Out Cross-Validation and Waic." *Journal of Statistcs and Computing* 27:5:

1413–32.

Venter, Gary, and Şule Şahin. 2018. "Parsimonious Parameterization of Age-Period-Cohort Models by Bayesian Shrinkage." *Astin Bulletin* 48:1: 89–110.