Obtaining Predictive Distributions for Reserves Which Incorporate Expert Opinion

by R. J. Verrall

ABSTRACT

This paper shows how expert opinion can be inserted into a stochastic framework for loss reserving. The reserving methods used are the chain-ladder and Bornhuetter-Ferguson, and the stochastic framework follows England and Verrall [8]. Although stochastic models have been studied, there are two main obstacles to their more frequent use in practice: ease of implementation and adaptability to user needs. This paper attempts to address these obstacles by utilizing Bayesian methods, and describing in some detail the implementation, using freely available software and programs supplied in the Appendix.

KEYWORDS

Bayesian statistics, Bornhuetter-Ferguson, chain-ladder, claims reserving, expert opinion, risk

1. Introduction

There has been a lot of attention given to stochastic reserving methods in the actuarial literature over recent years. Useful summaries can be found in England and Verrall [8] and Taylor [17]. The reader is strongly recommended to read England and Verrall [8], which contains more details on the basic models, before reading this paper.

There have been many useful things that have resulted from the recent papers on stochastic loss reserving: it is now possible to use a variety of methods to obtain reserve estimates, prediction intervals, predictive distributions, and so on. It is possible to use these methods for assessing the reserving risk, and for modeling a portfolio, line of business, or a whole company in a dynamic financial analysis. In short, the research published in recent years has been very successful in enhancing the understanding of loss reserving methods. This has been done by establishing stochastic approaches to models that are commonly used for loss reserving—for example, the chain-ladder technique, the Hoerl curve, and other parametric and non-parametric models. The stochastic approaches have added further models to the range of possible approaches. To take just one example, England and Verrall [7] showed how a nonparametric approach can be used to define a complete spectrum of models, with the chain-ladder technique at one end and the Hoerl curve at the other end.

In practical terms, it appears that the stochastic approaches that have found most popularity are those that are the simplest to implement. To pick out two examples, both Mack's model ([11]) and the bootstrap ([6] and [5]) are straightforward to implement in a spreadsheet. In contrast, using the full statistical model requires the use of statistical software, with some careful programming. It is not surprising, therefore, that a practitioner requiring prediction intervals as well as reserve estimates, or simply wanting to investigate the use of a stochastic approach, should choose the methods that are simplest to implement.

One aspect of reserving that has not, so far, received a great deal of attention in the literature is the question of intervention in the process by the actuary. In other words, the stochastic models have largely concentrated on providing a framework for the basic, standard methods. When these are used in practice, it is common to apply some expert knowledge or opinion to adjust the results before they are used. Examples of situations when intervention may be desirable is when there has been a change in the payment pattern due to a change in company policy, or where legislatures have enacted benefit limitations that restrict the potential for loss development and require an adjustment to historical development factors.

While it is possible to intervene in some models, the tendency is for this intervention to disrupt the assumptions made in the stochastic framework. For example, it is possible to change one or more of the residuals before applying a bootstrapping procedure, if the observed residuals appear to be out of line with what might be expected. But if this is done, the validity of the stochastic assumptions may be compromised. To take another example, consider the chain-ladder technique. This method involves the estimation of development factors, but it is often the case that these are adjusted before being applied to obtain reserve estimates. If this is done, the estimates from the stochastic model are being abandoned, and it is not clear what effect this might have on the prediction errors. For example, it is possible to calculate estimation errors for any parameter estimated in a stochastic model, but what estimation error should be used for a parameter that is simply inserted? The only way to address this properly is to use the Bayesian approach, and this provides an important motivation for the ideas discussed in this paper.

A second area where expert knowledge is applied is when the Bornhuetter-Ferguson [1] technique is used. This method uses the development factors from the chain-ladder technique, but it does not apply these to the latest cumulative losses to estimate the outstanding losses. Instead, an estimate is first procured separately, using background knowledge about the claims. This is then used with the development factors to obtain reserve estimates. Although not originally formulated using a Bayesian philosophy, the Bornhuetter-Ferguson technique is quite clearly suited to this approach because of the basic idea of what it is trying to do: incorporate expert opinion. Thus, we have a second important motivation for considering the use of Bayesian reserving methods. These are two very important examples of reserving approaches commonly used, which are best modeled using Bayesian methods. Among previous papers to discuss Bayesian loss reserving, we would mention de Alba [4] and Ntzoufras and Dellaportas [13].

One important property of Bayesian methods that makes them suitable for use with a stochastic reserving model is that they allow us to incorporate expert knowledge in a natural way, overcoming any difficulties about the effect on the assumptions made. In this paper, we consider the use of Bayesian models for loss reserving in order to incorporate expert opinion into the prediction of reserves. We concentrate on two areas as mentioned above: the Bornhuetter-Ferguson technique and the insertion of prior knowledge about individual development factors in the chain-ladder technique. The possibility of including expert knowledge is an important property of Bayesian models, but there is another equally important point: the ease with which they can be implemented. This is due to modern developments in Bayesian methodology based on so-called "Markov chain Monte Carlo" (MCMC) methods. It is difficult to emphasize enough the effect these methods have had on Bayesian statistics, but the books by Congdon ([3] and [2]) give some idea of the scope of the applications for which they have been used. The crucial aspect as far as this paper is concerned is that they are based on simulation, and therefore have some similarities with bootstrapping methods that, as was mentioned above, have gained in popularity for loss reserving.

It is also important that easy-to-use software is now available that allows us to implement the Bayesian models for loss reserving. While it is straightforward to define a Bayesian model, it is not always so easy to find the required posterior distributions for the parameters and predictive distributions for future observations. However, this has been made much easier in recent years by the development of MCMC methods, and by the software package winBUGS [16]. This software package is freely available from http://www.mrc-bsu.cam.ac.uk/bugs, and the programs for carrying out the Bayesian analysis for the models described in this paper are contained in the Appendix. Section 6.1 provides instructions on downloading this software. An excellent reference for actuarial applications of MCMC methods using winBUGS is Scollnik [15].

The basic idea behind MCMC methods is to simulate the posterior distribution by breaking the simulation process down into a number of simulations that are as easy to carry out as possible. This overcomes a common problem with Bayesian methods—that it can be difficult to derive the posterior distribution, which may in many cases be multidimensional. Instead of trying to simulate all the parameters at once, MCMC methods use the conditional distribution of each parameter, given all the others. In this way, the simulation is reduced to a univariate distribution, which is much easier to deal with. A Markov chain is formed because each parameter is considered in turn, and it is a simulation-based method: hence the term Markov chain Monte Carlo. For the readers for whom this is the first

time they have encountered MCMC methods, it is suggested that they simply accept that they are a neat way to get the posterior distributions for Bayesian models and continue reading the paper. If they like the ideas and would like to find out more, Scollnik [15] gives a much fuller account than is possible here, and the reader is advised to spend time working through some simpler examples with the help of the Scollnik paper.

This paper is set out as follows. In Section 2, we describe the notation and basic methods used, and in Section 3 we summarize the stochastic models used in the context of the chain-ladder technique. Sections 4 and 5 describe the Bayesian models for incorporating prior information into the reserving process. In Section 6 we describe in some detail how to implement the Bayesian models so that the reader can investigate the use of these models, using the programs given in the Appendix. In Section 7 we state some conclusions.

2. Notation and basic methods

To begin with, we define the notation used in this paper, and in doing so we briefly summarize the chain-ladder technique and the Bornhuetter-Ferguson method.

Although the methods can also be applied to other shapes of data, in order that the notation should not get too complicated we make the assumption that the data is in the shape of a triangle. Thus, without loss of generality, we assume that the data consist of a triangle of incremental losses:

$$\begin{array}{llll} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & \cdots & C_{2,n-1} & \\ \vdots & & & \\ C_{n1} & & & \end{array}.$$

This can also be written as $\{C_{ij} : j = 1,\ldots,n-i+1; i = 1,\ldots,n\}$, where n is the number of accident years. C_{ij} is used to denote incremental losses, and D_{ij} is used to denote the cumulative losses, defined by:

$$D_{ij} = \sum_{k=1}^{j} C_{ik}. \tag{2.1}$$

One of the methods considered in this paper is the chain-ladder technique, and the development factors $\{\lambda_j : j = 2,\ldots,n\}$. The usual estimates of the development factors from the standard chain-ladder technique are

$$\hat{\lambda}_j = \frac{\sum_{i=1}^{n-j+1} D_{ij}}{\sum_{i=1}^{n-j+1} D_{i,j-1}}. \tag{2.2}$$

Note that we only consider forecasting losses up to the latest development year (n) so far observed, and no tail factors are applied. It would be possible to extend this to allow a tail factor, using the same methods, but no specific modeling is carried out in this paper of the shape of the run-off beyond the latest development year. Thus, we refer to cumulative losses up to development year n, $D_{in} = \sum_{k=1}^{n} C_{ik}$, as "ultimate losses." For the chain-ladder technique, the estimate of outstanding losses is $D_{i,n-i+1}(\hat{\lambda}_{n-i+2} \cdot \hat{\lambda}_{n-i+3} \ldots \hat{\lambda}_n - 1)$.

The first case we consider is when these development factor estimates are not used for all rows. In other words, we consider the more general case where there is a separate development factor in each row, $\lambda_{i,j}$. The standard chain-ladder model sets $\lambda_{i,j} = \lambda_j$, for $i = 1,2,\ldots,n-j+1$; $j = 2,3,\ldots,n$, but we consider allowing the more general case where development factors can change from row to row. Section 4 describes the Bayesian approach to this, allowing expert knowledge to be used to set prior distributions for these parameters. In this way, we will be able to intervene in the estimation of the development factors, or else simply leave them for the standard chain-ladder model to estimate.

In Section 5 we consider the Bornhuetter-Ferguson method. This method uses the development factors from the chain-ladder technique, but

it incorporates knowledge about the "level" of each row by replacing the chain-ladder estimate of outstanding claims, $D_{i,n-i+1}(\hat{\lambda}_{n-i+2}\hat{\lambda}_{n-i+3}\cdots\hat{\lambda}_n - 1)$ by $M_i(1/(\hat{\lambda}_{n-i+2}\hat{\lambda}_{n-i+3}\cdots\hat{\lambda}_n))(\hat{\lambda}_{n-i+2}\cdot\hat{\lambda}_{n-i+3}\cdots\hat{\lambda}_n - 1)$. Here, M_i denotes a value for the ultimate losses for accident year i that is obtained using expert knowledge about the losses (for example, taken from the premium calculation). Thus, $M_i(1/(\hat{\lambda}_{n-i+2}\hat{\lambda}_{n-i+3}\cdots\hat{\lambda}_n))$ replaces the latest cumulative losses for accident year i, to which the usual chain-ladder parameters are applied to obtain the estimate of outstanding losses. From this, it can be seen that the difference between the Bornhuetter-Ferguson method and the chain-ladder technique is that the Bornhuetter-Ferguson technique uses an external estimate of the "level" of each row in the triangle, while the chain-ladder technique uses the data in that row itself. The Bornhuetter-Ferguson method can be formulated using a Bayesian approach, with the information about the external estimates for each row being used to form the prior distributions, as in Section 5.

This section has defined the notation used in the paper, and outlined the basic reserving methods that will be considered using stochastic approaches. In order to do this, a brief introduction to the stochastic models is needed, and this is given in Section 3.

3. Stochastic models for the chain-ladder technique

This section gives a brief summary of stochastic models that are related to the chain-ladder technique. A much fuller account may be found in England and Verrall [8], and in that paper's references and discussion. We consider the chain-ladder technique and note that it is possible to apply Bayesian methods in a similar way to other models.

There are a number of different approaches that can be taken to the chain-ladder technique, with various positivity constraints, all of which give the same reserve estimates as the chain-ladder technique. The connections between the chain-ladder technique and various stochastic models have been explored in a number of previous papers. For example, Mack [11] takes a non-parametric approach and specifies only the first two moments for the cumulative losses. In Mack's model the conditional mean and variance of $D_{ij} \mid D_{i,j-1}, \lambda_j, \sigma_j^2$ are $\lambda_j D_{i,j-1}$ and $\sigma_j^2 D_{i,j-1}$, respectively. Estimates of all the parameters are derived, and the properties of the model are examined. As was stated in the introduction, one of the advantages of this approach is that the parameter estimates and prediction errors can be obtained using a spreadsheet, without having recourse to a statistical package or any complex programming. The consequence of not specifying a distribution for the data is that there is no predictive distribution. Also, there are separate parameters in the variance that must also be estimated, separately from the estimation of the development factors.

As a separate stream of research, generalized linear models have also been considered. Renshaw and Verrall [14] used an approach based on generalized linear models [12] and examined the over-dispersed Poisson model for incremental losses:

> $C_{ij} \mid c, \alpha, \beta, \varphi \sim$ independent over-dispersed Poisson, with mean, m_{ij}, where $\log(m_{ij}) = c + \alpha_i + \beta_j$, and $\alpha_1 = \beta_1 = 0$.

The term "over-dispersed" requires some explanation. It is used here in connection with the Poisson distribution, and it means that if $X \sim \text{Poisson}(\mu)$, then $Y = \varphi X$ follows the over-dispersed Poisson distribution with $E(Y) = \varphi\mu$ and $V(Y) = \varphi^2 E(X) = \varphi^2\mu$. φ is usually greater than 1—hence the term "over-dispersed"—but this is not a necessity. It can also be used for other distributions, and we make use of it for the negative binomial distribution. As with the Poisson distribution, the over-dispersed negative binomial dis-

tribution is defined such that if $X \sim$ negative binomial then $Y = \varphi X$ follows the over-dispersed negative binomial distribution. Furthermore, a quasi-likelihood approach is taken so that the loss data are not restricted to the positive integers.

It can be seen that this formulation has some similarities with the model of Kremer [9], but it has a number of advantages. It does not necessarily break down if there are negative incremental loss values, it gives the same reserve estimates as the chain-ladder technique, and it has been found to be more stable than the log-normal model of Kremer. For these reasons, we concentrate on it in this paper. There are a number of ways of writing this model, which are useful in different contexts (note that the reserve estimates are unaffected by the way the model is written). In a strict sense, the formulation requires that the data are positive—otherwise it is more difficult to justify and interpret the inferences made from the data. However, in a purely practical context, it is useful to note that the estimation does not break down in the presence of some negative values.

Another way of writing the over-dispersed Poisson model for the chain-ladder technique is as follows:

$C_{ij} \mid x, y, \varphi \sim$ independent over-dispersed Poisson, with mean $x_i y_j$, and $\sum_{k=1}^{n} y_k = 1$.

Here $x = \{x_1, x_2, \ldots, x_n\}$ and $y = \{y_1, y_2, \ldots, y_n\}$ are parameter vectors relating to the rows (accident years) and columns (development years), respectively, of the run-off triangle. The parameter $x_i = E[D_{in}]$, and so represents expected ultimate cumulative losses (up to the latest development year so far observed, n) for the ith accident year. The column parameters, y_j, can be interpreted as the proportions of ultimate losses that emerge in each development year.

Although the over-dispersed Poisson models give the same reserve estimates as the chain-ladder technique (as long as the row and column sums of incremental claims are positive), the connection with the chain-ladder technique is not immediately apparent from this formulation of the model. For this reason, the negative binomial model was developed by Verrall [20], building on the over-dispersed Poisson model. Verrall showed that the same predictive distribution can be obtained from a negative binomial model (also with the inclusion of an over-dispersion parameter). In this recursive approach, the incremental claims have an over-dispersed negative binomial distribution, with mean and variance

$(\lambda_j - 1)D_{i,j-1}$ and $\varphi\lambda_j(\lambda_j - 1)D_{i,j-1}$, respectively.

Again, the reserve estimates are the same as the chain-ladder technique, and the same positivity constraints apply as for the over-dispersed Poisson model. It is clear from this that the column sums must be positive, since a negative sum would result in a development factor less than 1 ($\lambda_j < 1$), causing the variance to be negative. It is important to note that exactly the same predictive distribution can be obtained from either the Poisson or negative binomial models. Verrall [20] also argued that the model could be specified either for incremental or cumulative losses, with no difference in the results. The negative binomial model has the advantage that the form of the mean is exactly the same as that which naturally arises from the chain-ladder technique. In fact, by adding the previous cumulative losses, an equivalent model for $D_{ij} \mid D_{i,j-1}, \lambda_j, \varphi$ has an over-dispersed negative binomial distribution, with mean and variance

$\lambda_j D_{i,j-1}$ and $\varphi\lambda_j(\lambda_j - 1)D_{i,j-1}$, respectively.

Here the connection with the chain-ladder technique is immediately apparent because of the format of the mean.

Another model, which is not considered further in this paper, is closely connected with Mack's model, and deals with the problem of

negative incremental claims. This model replaces the negative binomial by a normal distribution, whose mean is unchanged, but whose variance is altered to accommodate the case when $\lambda_j < 1$. Preserving as much of $\lambda_j(\lambda_j - 1)D_{i,j-1}$ as possible, the variance is still proportional to $D_{i,j-1}$, with the constant of proportionality depending on j, but a normal approximation is used for the distribution of incremental claims. Thus, $C_{ij} \mid D_{i,j-1}, \lambda_j, \varphi_j$ is approximately normally distributed, with mean and variance

$$D_{i,j-1}(\lambda_j - 1) \text{ and } \varphi_j D_{i,j-1}, \text{ respectively,}$$

or $D_{ij} \mid D_{i,j-1}, \lambda_j, \varphi_j$ is approximately normally distributed, with mean and variance

$$\lambda_j D_{i,j-1} \text{ and } \varphi_j D_{i,j-1}, \text{ respectively.}$$

As for Mack's model, there is now another set of parameters in the variance that needs to be estimated.

For each of these models, the mean square error of prediction can be obtained, allowing the construction of prediction intervals, for example. Loss reserving is a predictive process: given the data, we try to predict future loss emergence. These models apply to all the data, both observed and future observations. The estimation is based on the observed data, and we require predictive distributions for the future observation.

We use the expected value of the distribution of future losses as the prediction. When considering variability, attention is focused on the root mean squared error of prediction (RMSEP), also known as the prediction error. To explain what this is, we consider, for simplicity, a random variable, y, and a predicted value $\hat{y}$. The mean squared error of prediction (MSEP) is the expected square difference between the actual outcome and the predicted value, $E[(y - \hat{y})^2]$, and can be written as follows:

$$E[(y - \hat{y})^2] = E[((y - E[y]) - (\hat{y} - E[y]))^2]. \tag{3.1}$$

In order to obtain an estimate of this, it is necessary to plug in $\hat{y}$ instead of y in the final expectation. Then the MSEP can be expanded as follows:

$$\begin{aligned} E[(y - \hat{y})^2] \approx\ & E[(y - E[y])^2] \\ & - 2E[(y - E[y])(\hat{y} - E[\hat{y}])] \\ & + E[(\hat{y} - E[\hat{y}])^2]. \end{aligned} \tag{3.2}$$

Assuming future observations are independent of past observations, the middle term is zero, and

$$E[(y - \hat{y})^2] \approx E[(y - E[y])^2] + E[(\hat{y} - E[\hat{y}])^2]. \tag{3.3}$$

In words, this is

prediction variance = process variance + estimation variance.

It is important to understand the difference between the prediction error and the standard error. Strictly, the standard error is the square root of the estimation variance. The prediction error is concerned with the variability of a forecast, taking account of uncertainty in parameter estimation and also of the inherent variability in the data being forecast. Further details of this can be found in England and Verrall [8].

Using non-Bayesian methods, these two components—the process variance and the estimation variance—are estimated separately, and Section 7 of England and Verrall [8] goes into detail about this. The direct calculation of these quantities can be a tricky process, and this is one of the reasons for the popularity of the bootstrap. The bootstrap uses a fairly simple simulation approach to obtain simulated estimates of the prediction variance in a spreadsheet. Fortunately, the same advantages apply to the Bayesian methods: the full predictive distribution can be found using simulation methods, and the RMSEP can be obtained directly by calculating its standard deviation. In addition, it is preferable to have the full predictive distribution, rather than just the

first two moments, which is another advantage of Bayesian methods.

The purpose of this paper is to show how expert opinion, from sources other than the specific data set under consideration, can be incorporated into the predictive distributions of the reserves. We use the approach of generalized linear models outlined in this section, concentrating on the over-dispersed Poisson and negative binomial models. We begin with considering how it is possible to intervene in the development factors for the chain-ladder technique in Section 4, and then consider the Bornhuetter-Ferguson method in Section 5.

4. Incorporating expert opinion about the development factors

In this section, the approach of Verrall and England [21] is used to show how to specify a Bayesian model that allows the practitioner to intervene in the estimation of the development factors for the chain-ladder technique. There are a number of ways in which this could be used, and we describe some possibilities in this section. It is expected that a practitioner would be able to extend these to cover situations that, although not specifically covered here, would also be useful. The cases considered here are the intervention in a development factor in a particular row, and the choice of how many years of data to use in the estimation. The reasons for intervening in these ways could be that there is information that the settlement pattern has changed, making it inappropriate to use the same development factor for each row.

For the first case, what may happen in practice is that a development factor in a particular row is simply changed. Thus, although the same development parameters (and hence run-off pattern) are usually applied for all accident years, if there is some exogenous information that indicates that this is not appropriate, the practitioner may decide to apply a different development factor (or set of factors) in some, or all, rows.

In the second case, it is common to look at, say, five-year volume weighted averages in calculating the development factors, rather than using all the available data in the triangle. The Bayesian methods make this particularly easy to do and are flexible enough to allow many possibilities.

We use the negative binomial model described in Section 3, with different development factors in each row. This is the model for the data, and we then specify prior distributions for the development factors. In this way, we can choose prior distributions that reproduce the chain-ladder results, or we can intervene and use prior distributions based on external knowledge. The model for incremental claims, $C_{ij} \mid D_{i,j-1}, \lambda_{i,j}, \varphi$, is an over-dispersed negative binomial distribution, with mean and variance

$$(\lambda_{i,j} - 1)D_{i,j-1} \text{ and } \varphi\lambda_{i,j}(\lambda_{i,j} - 1)D_{i,j-1}, \text{ respectively.}$$

We next need to define prior distributions for the development factors, $\lambda_{i,j}$. It is possible to set some of these equal to each other (within each column) in order to revert to the standard chain-ladder model. This is done by setting

$$\lambda_{i,j} = \lambda_j \qquad \text{for} \quad i = 1,2,\ldots,n-j+1; \quad j = 2,3,\ldots,n$$

and defining vague prior distributions for λ_j $(j = 2,3,\ldots,n)$. This was the approach taken in Section 8.4 of England and Verrall [8] and is very similar to that taken by de Alba [4]. This can provide a very straightforward method to obtain prediction errors and predictive distributions for the chain-ladder technique.

However, we really want to move away from the basic chain-ladder technique, and construct Bayesian prior distributions that encompass the expert opinion about the development parameters. Suppose, for example, that we have a 10×10 triangle. We consider the two possibilities for incorporating expert knowledge described above.

To illustrate the first case, suppose that there is information that implies that the second development factor (from Column 2 to Column 3) should be given the value 1.5, for rows 8, 9, and 10, and that there is no indication that the other parameters should be treated differently from the standard chain-ladder technique. An appropriate way to treat this would be to specify

$$\lambda_{i,j} = \lambda_j \qquad \text{for} \quad i = 1,2,\ldots,n-j+1; \quad j = 2,4,5,\ldots,n$$

$$\lambda_{i,3} = \lambda_3 \qquad \text{for} \quad i = 1,2,\ldots,7$$

$$\lambda_{8,3} = \lambda_{9,3} = \lambda_{10,3}.$$

The means and variances of the prior distributions of the parameters are chosen to reflect the expert opinion:

$\lambda_{8,3}$ has a prior distribution with mean 1.5 and variance W, where W is set to reflect the strength of the prior information.

λ_j have prior distributions with large variances.

For the second case, we divide the data into two parts using the prior distributions. To do this, we set

$$\lambda_{i,j} = \lambda_j \qquad \text{for} \quad i = n-j-3, n-j-2, n-j-1, n-j, n-j+1$$

$$\lambda_{i,j} = \lambda_j^* \qquad \text{for} \quad i = 1,2,\ldots,n-j-4$$

and give both λ_j and λ_j^* prior distributions with large variances so that they are estimated from the data. Adjustments to the specification are made in the later development years, where there are less than five rows. For these columns there is just one development parameter, λ_j.

The specific form of the prior distribution (gamma, log-normal, etc.) is usually chosen so that the numerical procedures in winBUGS work as well as possible.

These models are used as illustrations of the possibilities for incorporating expert knowledge about the development pattern, but it is (of course) possible to specify many other prior distributions. In the Appendix, the winBUGS code is supplied, which can be cut and pasted directly in order to examine these methods. Section 6 contains a number of examples, including the ones described in this section.

Table 1. Parameters, mean and variance of a gamma distribution

α_i	β_i	M_i	M_i/β_i
10000	10	1000	100
1000	1	1000	1000
100	0.1	1000	10000

5. A Bayesian model for the Bornhuetter-Ferguson method

In this section, we show how the Bornhuetter-Ferguson method can be considered in a Bayesian context, using the approach of Verrall [19]. For further background on the Bornhuetter-Ferguson method, see Mack [10].

In Section 3, the over-dispersed Poisson model was defined as follows.

$C_{ij} \mid x, y, \varphi \sim$ independent over-dispersed Poisson, with mean $x_i y_j$, and $\sum_{k=1}^{n} y_k = 1$.

In the Bayesian context, we also require prior distributions for the parameters. The Bornhuetter-Ferguson method assumes that there is expert opinion about the level of each row, and we therefore concentrate first on the specification of prior distributions for these. The most convenient form to use is gamma distributions:

$$x_i \mid \alpha_i, \beta_i \sim \text{ independent } \Gamma(\alpha_i, \beta_i). \quad (5.1)$$

There is a wide range of possible choices for the parameters of these prior distributions, α_i and β_i. It is easiest to consider the mean and variance of the gamma distribution, α_i/β_i and α_i/β_i^2, respectively. These can be written as M_i and M_i/β_i, from which it can be seen that, for a given choice of M_i, the variance can be altered by changing the value of β_i. To consider a simple example, suppose it has been decided that $M_i = 1000$. Table 1 shows how the value of β_i affects the variance of the prior distribution, while M_i is kept constant.

Clearly, choosing a larger value of β_i implies we are more sure about the value of M_i, and choosing a smaller value means we are less sure.

We now consider the effect of using these prior distributions on the model for the data. Recall that, for the chain-ladder technique, the mean of the distribution of incremental claims may be written as $(\lambda_j - 1)D_{i,j-1}$.

Using a similar approach, Verrall [20] and Verrall [19] derive the distribution of C_{ij}, given the past data, after the row parameters have been estimated. In a Bayesian context, this means first deriving the posterior distribution of the row parameters given the data using a standard prior-posterior analysis:

$$f(x_i \mid y; data) \propto f(data \mid x, y) f(x_i \mid \alpha_i, \beta_i). \tag{5.2}$$

Note that, if we are considering C_{ij}, the *data* used here is $C_{i1}, C_{i2}, \ldots, C_{i,j-1}$. Having obtained this distribution, the distribution of the next observation can be found as follows:

$$f(C_{ij} \mid y; data) = \int f(C_{ij} \mid x_i, y) f(x_i \mid y; data) dx_i. \tag{5.3}$$

This result is derived in detail in Verrall [19], where it is shown that it is possible to rewrite it in terms of the usual chain-ladder development factors, λ_j, rather than using the column parameters y_j. For full details of the derivation, the reader is referred to Verrall [19]. For the purposes of this paper, the important point is that the mean of C_{ij} for the Bayesian model is

$$Z_{ij}(\lambda_j - 1)D_{i,j-1} + (1 - Z_{ij})(\lambda_j - 1)M_i \frac{1}{\lambda_j \lambda_{j+1} \ldots \lambda_n},$$

where

$$Z_{ij} = \frac{\sum_{k=1}^{j-1} y_k}{\beta_i \varphi + \sum_{k=1}^{j-1} y_k}.$$

This can be seen to be in the form of a credibility formula, and is a trade-off between the chain-ladder $((\lambda_j - 1)D_{i,j-1})$ and the Bornhuetter-Ferguson $((\lambda_j - 1)M_i(1/(\lambda_j \lambda_{j+1} \ldots \lambda_n)))$. The credibility factor, Z_{ij}, governs the trade-off between the prior mean and the data. We can influence the balance of this trade-off through the choice of β_i. In line with the discussion above, the larger the value of β_i the closer we get to the Bornhuetter-Ferguson method, and the smaller the value of β_i, the closer we get to the chain-ladder technique. In this way, we can use different specifications of the prior distributions for the row parameters in order to use the chain-ladder technique, the Bornhuetter-Ferguson method, or a complete spectrum of methods between these two extremes. If we choose to use prior distributions with large variances, we do not influence the parameter estimates and the result will be the same as (or extremely close to) the chain-ladder technique. If we use very small variances, we are saying that we are very sure what the parameter values should be and the results will be the same as (or very close to) the Bornhuetter-Ferguson method. Thus, we can use these methods within a stochastic framework, and we can also consider using the whole range of models that lie between these two.

We have yet to consider the estimation of the column parameters, other than to point out that the Bornhuetter-Ferguson method, being deterministic, simply plugs in the chain-ladder parameter estimates. We now consider this issue in more detail and define a Bayesian approach to the Bornhuetter-Ferguson method. One option is to simply use plug-in estimates, obtained, for example, from the straightforward chain-ladder technique. This is the approach used in the deterministic application of the Bornhuetter-Ferguson method, but it is not suitable here since we would prefer a stochastic approach. A better option is to define improper prior distributions for the column parameters, and estimate the column parameters *first*, before applying prior distributions for the row parameters and estimating these. This second option allows us to take into account the fact that the column parameters have been estimated when calculating the prediction errors, predictive distribution, etc. It is not required to

include any information about the column parameters, and hence we use improper gamma distributions for the column parameters, and derive the posterior distributions of these using a standard Bayesian prior-posterior analysis. The result of this is a distribution that looks similar to the negative binomial model for the chain-ladder technique, but which is recursive in i instead of j:

$C_{ij} \mid C_{1,j}, C_{2,j}, \ldots, C_{i-1,j}, x, \varphi \sim$ over-dispersed negative binomial, with mean $(\gamma_i - 1)\sum_{m=1}^{i-1} C_{m,j}$.

Comparing this to the mean of the chain-ladder model, $(\lambda_j - 1)D_{i,j-1} = (\lambda_j - 1)\sum_{m=1}^{j-1} C_{i,m}$, it can be seen that they are identical in form, with the recursion either being across the rows or down the columns.

In the context of the Bornhuetter-Ferguson method, we now have the stochastic version of this model. The Bornhuetter-Ferguson method inserts values for the expected ultimate claims in each row, x_i, in the form of the values M_i. In the Bayesian context, prior distributions will be defined for the parameters x_i, as discussed above. However, the model has been reparameterized, with a new set of parameters, γ_i. Hence, it is necessary to define the relationship between the new parameters, γ_i, and the original parameters, x_i. This is given in the equations below, which can be used to find values of γ_i from the values of x_i given in the prior distributions. Note that there was an error in the equation given in Verrall [19], and I am grateful to Peter England for pointing this out.

The Bornhuetter-Ferguson technique can be reproduced by using strong prior information for the row parameters, x, and the chain-ladder technique can be reproduced by using improper priors for the row parameters. In other words, the Bornhuetter-Ferguson technique assumes that we are completely sure about the values of the row parameters, and their prior distributions have very small variances, while the chain-ladder technique assumes there is no information and has very large variances.

The preceding equations have now defined a stochastic version of the Bornhuetter-Ferguson technique. Since the column parameters (the development factors) are dealt with first, using improper prior distributions, their estimates will be those implied by the chain-ladder technique. Prior information can be defined in terms of distributions for the parameters x_i, which can then be converted into values for the parameters γ_i, and this is implemented in Section 6.

6. Implementation

This section explains how the Bayesian models can be implemented, using the software package winBUGS [16] which is available from http://www.mrc-bsu.cam.ac.uk/bugs. The programs used in these illustrations are contained in the Appendix.

The data set used in this section is taken from Taylor and Ashe [18], and has also been used in a number of previous papers on stochastic reserv-

$$\begin{aligned}
\gamma_1 &= 1 \\
\gamma_2 &= 1 + \frac{x_2\left(1 - \frac{1}{\lambda_n}\right)}{C_{1n}} \\
\gamma_i &= 1 + \frac{x_i\left(1 - \frac{1}{\sum_{k=n-i+2}^{n}\lambda_k}\right)}{\sum_{m=1}^{i-1} C_{m,n} + \sum_{k=n-i+3}^{n}\left[\left(\prod_{l=n-k+2}^{i-1}\gamma_l\right)\sum_{m=1}^{n-k+1} C_{m,k}\right]} \qquad i = 3,\ldots,n.
\end{aligned} \tag{5.4}$$

Table 2. Data from Taylor and Ashe [18] with the chain-ladder estimates

Accident Year	Development Year									
	1	2	3	4	5	6	7	8	9	10
1	357,848	766,940	610,542	482,940	527,326	574,398	146,342	139,950	227,229	67,948
2	352,118	884,021	933,894	1,183,289	445,745	320,996	527,804	266,172	425,046	
3	290,507	1,001,799	926,219	1,016,654	750,816	146,923	495,992	280,405		
4	310,608	1,108,250	776,189	1,562,400	272,482	352,053	206,286			
5	443,160	693,190	991,983	769,488	504,851	470,639				
6	396,132	937,085	847,498	805,037	705,960					
7	440,832	847,631	1,131,398	1,063,269						
8	359,480	1,061,648	1,443,370							
9	376,686	986,608								
10	344,014									
Chain-ladder development factors:										
3.4906	1.7473	1.4574	1.1739	1.1038	1.0863	1.0539	1.0766	1.0177		
Chain-ladder reserve estimates:										
2	94,634									
3	469,511									
4	709,638									
5	984,889									
6	1,419,459									
7	2,177,641									
8	3,920,301									
9	4,278,972									
10	4,625,811									
Overall	18,680,856									

ing. The incremental loss data is given in Table 2, together with the chain-ladder results for comparison purposes.

Before looking at the uses of the Bayesian models, we should discuss the nuisance parameter φ. In a full Bayesian analysis, we should also give this a prior distribution and estimate it along with the other parameters. However, for ease of implementation we instead use a plug-in estimate, in line with the approach taken in classical methods (in England and Verrall [8], for example). The value used is that obtained from the straightforward application of the over-dispersed Poisson model, estimating the row and column parameters using maximum likelihood estimation (it is possible to use S-Plus or Excel for this).

6.1. Using the software

Before considering the results from the programs in any detail, we first describe how to set up the software and run one of the programs from scratch. An excellent reference in the context of actuarial modeling is Skollnik [15]. Table 2 shows the standard chain-ladder results, and in this section we will implement the model described in Section 5, but use the assumptions of the chain-ladder technique, rather than the Bornhuetter-Ferguson method. This means that we will use large variances for the prior distributions for the ultimate claims in each row, implying that there is no prior knowledge about them, and hence the results we obtain should be close to the chain-ladder results. Thus, we will first reproduce the results that can also be obtained using non-Bayesian methods (see England and Verrall [8] for more details of the non-Bayesian methods). After going through this example in detail, the remainder of this section will show how the Bayesian models incorporating prior knowledge described in Sections 4 and 5 can be imple-

mented, and illustrate the effect that the choice of prior distributions can have.

The steps necessary for implementing the chain-ladder technique in winBUGS are listed below.

1. Go to the web site, download the latest version of the software and install.
2. Go back to the web site and register, and you will be sent a copy of the key to unlock the software. Follow the instructions in the email for unlocking the software.
3. Once you have a fully functioning version of winBUGS, you can run the programs in the Appendix. Open winBUGS and click on "File" in the top toolbar, and then "New" in the pop-down list. This will open a new window.
4. Copy the program in (i) of the Appendix, including the word "model" at the top and all the data at the bottom, right down to where the next subsection begins at (ii). The last line is 0,0,0,0,0,0,0,0,0,0)). Paste all of this into the new window in winBUGS.
5. In winBUGS, select "Model" in the toolbar at the top and "Specification" in the pop-down list. This opens a new window called "Specification Tool."
6. Highlight the word "model" at the top of the program, and then click "check model" in the Specification Tool window. If all is well, it will say "model is syntactically correct" in the bottom left corner.
7. Now move down in the window containing the program until you get to #DATA. Highlight the word "list" immediately below that, and click "load data" in the Specification Tool window. It should say "data loaded" in the bottom left corner.
8. Click "compile" in the Specification Tool window. After a few seconds, it should say "model compiled" in the bottom left corner.
9. Now move down in the window containing the program until you get to #INITIAL VALUES. Highlight the word "list" immediately below that, and click "load inits" in the Specification Tool window. It should say "model is initialised" in the bottom left corner.
10. Select "Model" in the toolbar at the top and "Update" in the pop-down list. This opens a new window called "Update Tool." The number of iterations in the simulation process can be changed in this window by changing the figure next to "updates." Just at the moment, 1,000 is sufficient, so click on "update." This runs 1,000 simulations without storing the results. This may take a few minutes: don't be concerned if nothing appears to be happening! When it is complete, a message appears in the bottom left corner saying how long the updates took (for my laptop it was 221 seconds).
11. Select "Inference" in the toolbar at the top and "Samples" in the pop-down list. This opens a new window called "Sample Monitor Tool." We want to look at the row totals and overall total, which have been defined as a vector R and Total in the program. In the Sample Monitor Tool window, click in the box to the right of the word "node" and type R. Then click on "set." Repeat for Total, noting that it is case sensitive.
12. Return to the Update Tool Window and click on Update to perform 1,000 simulations. This should be quicker (6 seconds for my laptop). This time the values of R and Total will be stored.
13. Return to the Sample Monitor Tool window, type * in the box to the right of the word "node," and click "stats." This will give a new window with something like the results below. This completes the steps necessary for fitting the Bayesian model.

Table 3. Results

Node	Mean	sd	MC Error	2.5%	Median	97.5%	Start	Sample
R[2]	92750.0	110600.0	2963.0	779.2	56320.0	412800.0	1001	1000
R[3]	473900.0	223100.0	6424.0	1.52E+5	4.4E+5	1.011E+6	1001	1000
R[4]	7.05E+5	2.58E+5	9085.0	307600.0	674500.0	1.288E+6	1001	1000
R[5]	985800.0	304600.0	8127.0	467600.0	960600.0	1.667E+6	1001	1000
R[6]	1.417E+6	378300.0	13430.0	768500.0	1.399E+6	2.217E+6	1001	1000
R[7]	2.174E+6	5.19E+5	16850.0	1.271E+6	2.132E+6	3.233E+6	1001	1000
R[8]	3.925E+6	776900.0	28100.0	2.585E+6	3.885E+6	5.555E+6	1001	1000
R[9]	4.284E+6	1.066E+6	36840.0	2.464E+6	4.207E+6	6.731E+6	1001	1000
R[10]	4.641E+6	2.002E+6	61630.0	1.73E+6	4.407E+6	9.345E+6	1001	1000
Total	1.87E+7	3.056E+6	101600.0	1.314E+7	1.861E+7	2.554E+7	1001	1000

The columns of Table 3 headed "mean" and "sd" give the predicted reserves and prediction errors, and these values can be compared with the chain-ladder results in Table 2. Since this is a simulation process, the results will depend on the prior distributions, the initial values, and the number of iterations carried out. The prior distributions in the program had reasonably large variances, so the results should be close to the chain-ladder results. More simulations should be used in steps 10 and 12 (we use 10,000 in the illustrations below), and the prior variances could be increased. Using this number of simulations gives the results shown in Table 4.

The results certainly confirm that we can reproduce the chain-ladder results, and produce the prediction errors. It is also possible to obtain other information about the model from winBUGS. For example, it is possible to produce full predictive distributions, using "density" in the Sample Monitor Tool window.

We have now described one implementation of a Bayesian model using winBUGS. In the rest of this section, we consider the Bayesian models described in Sections 4 and 5 in order to consider how expert opinion can be incorporated into the predictive distribution of reserves. In each case, the programs are available in the Appendix, and the results can be reproduced using steps 3 to 13, above. It should be noted that this is a simulation-based program, so the results obtained may not exactly match the results given below. However, there should be no significant differences, with the differences that there are being due to simulation error.

Table 4. Chain-ladder results. the prediction error is equal to the Bayesian standard deviation

	Chain-Ladder Reserve	Bayesian Mean	Bayesian Standard Deviation	Prediction Error (%)
Year 2	94,634	94,440	111,100	118%
Year 3	469,511	471,400	219,400	47%
Year 4	709,638	716,300	263,600	37%
Year 5	984,889	991,600	308,100	31%
Year 6	1,419,459	1,424,000	374,700	26%
Year 7	2,177,641	2,186,000	497,200	23%
Year 8	3,920,301	3,935,000	791,000	20%
Year 9	4,278,972	4,315,000	1,068,000	25%
Year 10	4,625,811	4,671,000	2,013,000	43%
Overall	18,680,856	18,800,000	2,975,000	16%

6.2. Intervention in the chain-ladder technique

We now consider using a prior distribution to intervene in some of the parameters of the chain-ladder model, instead of using prior distributions with large variances that just reproduce the chain-ladder estimates. The implementation is set up in Section (ii) of the Appendix, and the program can be cut and pasted into winBUGS and run following steps 3 onwards, above.

We consider two cases, as discussed in Section 4. For the first case, we assume that there is information that implies that the second develop-

Table 5. Individual development factors

Accident Year	Development Year								
	2	3	4	5	6	7	8	9	10
1	3.143	1.543	1.278	1.238	1.209	1.044	1.04	1.063	1.018
2	3.511	1.755	1.545	1.133	1.084	1.128	1.057	1.086	
3	4.448	1.717	1.458	1.232	1.037	1.12	1.061		
4	4.568	1.547	1.712	1.073	1.087	1.047			
5	2.564	1.873	1.362	1.174	1.138				
6	3.366	1.636	1.369	1.236					
7	2.923	*1.878*	1.439						
8	3.953	*2.016*							
9	3.619								

ment factor (from Column 2 to Column 3) should be given the value 1.5 for rows 7, 8, 9, and 10, and that there is no indication that the other parameters should be treated differently from the standard chain-ladder technique. In order to implement this, the parameter for the second development factor for rows 7–10 is given a prior distribution with mean 1.5. We then look at two different choices for the prior variance for this parameter. Using a large variance means that the parameter is estimated separately from the other rows, but using the data without letting the prior mean influence it too greatly. We then use a standard deviation of 0.1 for the prior distribution, so that the prior mean has a greater influence.

We consider first the estimate of the second development factor. The chain ladder estimate is 1.7473 and the individual development factors for the triangle are shown in Table 5. The rows for the second development factor that are modeled separately are shown in italics. The estimate using the Bayesian models is 1.68 for rows 1–6. When a large variance is used for the prior distribution of the development factor for rows 7–10, the estimate using the Bayesian model is 1.971. With the smaller variance for this prior distribution, the estimate is 1.673, and has been drawn down towards the prior mean of 1.5. This clearly shows how the prior distributions can be used to influence the parameter estimates.

The effect on the reserve estimates is shown in Table 6, which compares the reserves and prediction errors for the two cases outlined above with the results for the chain-ladder model (which could be produced using the program in Section 6.1 on this set of data). The chain-ladder figures are slightly different from those given in Table 4 because this is a simulation method.

It is interesting to note that, in this case, the intervention has not had a marked effect on the prediction errors (in percentage terms). However, the prediction errors themselves have changed considerably, and this indicates that it is important to think of the prediction error as a percentage of the prediction. Other prior distributions could have a greater effect on the percentage prediction error.

The second case we consider is when we use only the most recent data for the estimation of each development factor. For the last three development factors, all the data is used because there is no more than three years for each. For the other development factors, only the three most recent years are used. The estimates of the development factors are shown in Table 7. The estimates of the first development factor are not affected by the change in the model (the small differences could be due to simulation error or the changes elsewhere). For the other development factors, the estimates can be seen to be affected by the model assumptions.

The effect of using only the latest three years in the estimation of the development factors in

Table 6. Reserves and prediction errors for the chain-ladder and Bayesian models

	Chain-Ladder		Large Variance		Small Variance	
	Reserve	Prediction Error	Reserve	Prediction Error	Reserve	Prediction Error
Year 2	97,910	115%	95,920	116%	95,380	117%
Year 3	471,200	46%	475,700	46%	470,500	47%
Year 4	711,100	38%	721,700	37%	714,400	37%
Year 5	989,200	31%	996,800	31%	994,700	31%
Year 6	1,424,000	27%	1,429,000	26%	1,428,000	27%
Year 7	2,187,000	23%	2,196,000	23%	2,185,000	23%
Year 8	3,930,000	20%	3,937,000	20%	3,932,000	20%
Year 9	4,307,000	24%	4,998,000	27%	4,044,000	25%
Year 10	4,674,000	43%	5,337,000	44%	4,496,000	43%
Overall	18,790,000	16%	20,190,000	17%	18,360,000	16%

Table 7. Development factors using three most recent years' data separately

Accident Year	Development Year								
	2	3	4	5	6	7	8	9	10
1	3.143	1.543	1.278	1.238	1.209	1.044	1.04	1.063	1.018
2	3.511	1.755	1.545	1.133	1.084	1.128	1.057	1.086	
3	4.448	1.717	1.458	1.232	1.037	1.12	1.061		
4	4.568	1.547	1.712	1.073	1.087	1.047			
5	2.564	1.873	1.362	1.174	1.138				
6	3.366	1.636	1.369	1.236					
7	2.923	1.878	1.439						
8	3.953	2.016							
9	3.619								
Earlier rows	3.575	1.688	1.513	1.197	1.139	1.045			
Recent rows	3.579	1.852	1.393	1.155	1.085	1.099	1.054	1.076	1.018
All rows	3.527	1.751	1.46	1.175	1.104	1.087	1.054	1.076	1.018

the forecasting of outstanding claims can be seen in Table 8.

In this case, the effect on the reserves is not particularly great. The prediction errors have increased for most years, although the effect is not great on these either. The importance of the Bayesian method is to actually be able to assess the effect of using different sets of data on the uncertainty of the outcome.

6.3. The Bornhuetter-Ferguson method

In this section, we consider intervention on the level of each row, using the Bornhuetter-Ferguson method. We consider two examples. The first uses small variances for the prior distributions of the row parameters, thus reproducing the Bornhuetter-Ferguson method. The second example uses less strong prior information, and

Table 8. Reserve estimates using three most recent years' data

	Chain-Ladder		Bayesian Model	
	Reserve	Prediction Error	Reserve	Prediction Error
Year 2	97,910	115%	94,860	115%
Year 3	471,200	46%	469,300	46%
Year 4	711,100	38%	712,900	37%
Year 5	989,200	31%	1,042,000	30%
Year 6	1,424,000	27%	1,393,000	27%
Year 7	2,187,000	23%	2,058,000	24%
Year 8	3,930,000	20%	3,468,000	22%
Year 9	4,307,000	24%	4,230,000	27%
Year 10	4,674,000	43%	4,711,000	47%
Overall	18,790,000	16%	18,180,000	18%

produces results that lie between the Bornhuetter-Ferguson method and the chain-ladder technique. We use the negative binomial model for the data that was described in Section 5, and the winBUGS code for this is given in the Appendix,

Table 9. Negative binomial model: Bayesian model with precise priors for all rows: mean and prediction error of reserves

	Bayesian Mean Reserve	Bayesian Prediction Error	Bayesian Prediction Error %	Bornhuetter-Ferguson Reserve
Year 2	95,680	111,100	116%	95,788
Year 3	482,500	211,900	44%	480,088
Year 4	736,400	250,100	34%	736,708
Year 5	1,118,000	296,500	27%	1,114,999
Year 6	1,533,000	339,700	22%	1,527,444
Year 7	2,305,000	410,300	18%	2,308,139
Year 8	3,474,000	497,500	14%	3,466,839
Year 9	4,547,000	555,000	12%	4,550,270
Year 10	5,587,000	610,900	11%	5,584,677
Overall	19,880,000	1,854,000	9%	19,864,951

section (i). Section 6.1 used this method with large variances for the prior, thereby reproducing the chain-ladder technique.

First we consider the Bornhuetter-Ferguson method, exactly as it is usually applied. For this, we begin by using prior distributions for the row parameters, which all have standard deviation 1,000 (which is small compared with the means), and whose means are:

x_2	x_3	x_4	x_5
5,500,000	5,500,000	5,500,000	5,500,000
x_6	x_7	x_8	x_9
5,500,000	6,000,000	6,000,000	6,000,000
x_{10}			
6,000,000			

In order to implement this, using the code in the Appendix, it is necessary to change the "DATA" section of the program (just before the "INITIAL VALUES" section). It is explained in the program exactly what changes to make.

The Bornhuetter-Ferguson estimates of outstanding losses, and the results from the Bayesian model are shown in Table 9.

In this case, it can be seen that the results are very close to those of the Bornhuetter-Ferguson technique. Thus, if it is desired to use the Bornhuetter-Ferguson method within this stochastic framework, this is the approach that should be used. The added information available is the prediction errors. Further, it is possible to generate predictive distributions rather than just the mean and prediction error.

The Bornhuetter-Ferguson technique assumes that there is strong prior information about the row parameters, so that the standard deviations of the prior distributions used in this example are small. The other end of the spectrum is constituted by the chain-ladder technique, when large standard deviations are used for the prior distributions. Between these two extremes is a whole range of possible models, which can be specified by using different standard deviations. We now illustrate the results when less strongly informative prior distributions are used for the row parameters. We use the same prior means as above, but this time use a standard deviation of 1,000,000. We are incorporating prior belief about the ultimate losses for each year, but allowing for uncertainty in this information. The associated reserve results are shown in Table 10. Notice that the reserves are between the chain-ladder and Bornhuetter-Ferguson results. Notice also that the precision of the prior has influenced the prediction errors, but to a lesser extent. This provides an extra level of flexibility, allowing for the choice of a range of models in a continuous spectrum between the chain-ladder technique and Bornhuetter-Ferguson.

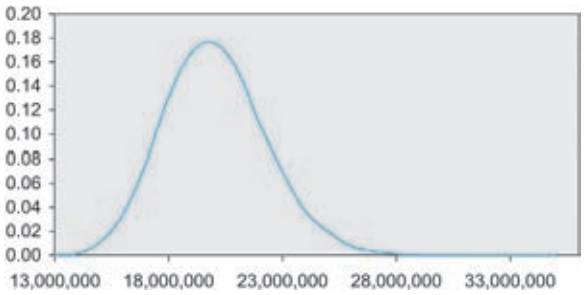
7. Conclusions

This paper has shown how expert opinion, separate from the reserving data, can be incorporated into the prediction intervals for a stochastic model. The advantages of a stochastic approach are that statistics associated with the predictive distribution are also available, rather than just a point estimate. In fact, it is possible to produce the full predictive distribution, rather than just

Table 10. Negative binomial model: Bayesian model with informative priors: mean and prediction error of reserves

	Bayesian Mean Reserve	Bayesian Prediction Error	Bayesian Prediction Error	Bornhuetter-Ferguson Reserve	Chain-Ladder Reserve
Year 2	94,660	111,500	118%	95,788	94,634
Year 3	470,400	218,800	47%	480,088	469,511
Year 4	717,100	265,900	37%	736,708	709,638
Year 5	994,900	308,900	31%	1,114,999	984,889
Year 6	1,431,000	376,800	26%	1,527,444	1,419,459
Year 7	2,198,000	488,900	22%	2,308,139	2,177,641
Year 8	3,839,000	727,200	19%	3,466,839	3,920,301
Year 9	4,417,000	865,500	20%	4,550,270	4,278,972
Year 10	5,390,000	1,080,000	20%	5,584,677	4,625,811
Overall	19,550,000	2,252,000	12%	19,864,951	18,680,856

Figure 1. Distribution of reserve for Bornhuetter-Ferguson estimation



the first two moments. As was emphasized by England and Verrall [8], the full predictive distribution contains a lot more information than just its mean and standard deviation, and it is a great advantage to be able to look at this distribution. As an illustration of this, Figure 1 shows the predictive distribution of outstanding losses for the final example considered above, in Section 6.3, Table 10.

A further possibility for including expert knowledge within a stochastic framework applies when the Bornhuetter-Ferguson technique is used. This is an adaptation of the method used in Sections 5 and 6.3, whereby the reserve is specified rather than the ultimate losses, u_i. The reserve value can be used to infer a value for u_i, from which the stochastic version of the Bornhetter-Ferguson method can be applied.

We have concentrated on two important situations that we believe are most common when expert opinion is used. However, the same approach could also be taken in other situations and for other modeling methods, such as the Hoerl curve. This would allow us to add tail factors to the models considered in this paper. This paper has been more concerned with the general approach rather than specific reserving methods. However, we acknowledge that methods based on the chain-ladder setup are very commonly used and we hope that, by using this framework, we enable actuaries to appreciate the suggestions made in this paper, and to experiment with the programs supplied.

References

[1] Bornhuetter, R. L. and R. E. Ferguson, "The Actuary and IBNR," *Proceedings of the Casualty Actuarial Society* 59, 1972, pp. 181–195.

[2] Congdon, Peter, *Applied Bayesian Modelling*, Somerset, NJ: Wiley, 2001.

[3] Congdon, Peter, *Bayesian Statistical Modelling*, Somerset, NJ: Wiley, 2001.

[4] de Alba, Enrique, "Bayesian Estimation of Outstanding Claims Reserves," *North American Actuarial Journal* 6:4, 2002, pp. 1–20.

[5] England, P. D., Addendum to "Analytic and Bootstrap Estimates of Prediction Errors in Claims Reserving," *Insurance: Mathematics and Economics* 31, 2002, pp. 461–466.

[6] England, P. D. and R. J. Verrall, "Analytic and Bootstrap Estimates of Prediction Errors in Claims Reserving," *Insurance: Mathematics and Economics* 25, 1999, pp. 281–293.

[7] England, P. D. and R. J. Verrall, "A Flexible Framework for Stochastic Claims Reserving," *Proceedings of the Casualty Actuarial Society* 88, 2001, pp. 1–38.
[8] England, P. D. and R. J. Verrall, "Stochastic Claims Reserving in General Insurance" (with discussion), *British Actuarial Journal* 8, 2002, pp. 443–554.
[9] Kremer, E., "IBNR Claims and the Two Way Model of ANOVA," *Scandinavian Actuarial Journal* 1, 1982, pp. 47–55.
[10] Mack, T., "Credible Claims Reserves: The Benktander Method," *ASTIN Bulletin* 30, 2000, pp. 333–347.
[11] Mack, T., "Distribution-Free Calculation of the Standard Error of Chain-Ladder Reserve Estimates," *ASTIN Bulletin* 23, 1993, pp. 213–225.
[12] McCullagh, P. and J. Nelder. *Generalized Linear Models* (2nd edition), London: Chapman and Hall, 1989.
[13] Ntzoufras, I. and P. Dellaportas, "Bayesian Modelling of Outstanding Liabilities Incorporating Claim Count Uncertainty," *North American Actuarial Journal* 6:1, 2002, pp. 113–128.
[14] Renshaw, A. E. and R. J. Verrall, "A Stochastic Model Underlying the Chain-Ladder Technique," *British Actuarial Journal* 4, 1998, pp. 903–923.
[15] Scollnik, D. P. M., "Actuarial Modeling with MCMC and BUGS," *North American Actuarial Journal* 5:2, 2001, pp. 96–124.
[16] Spiegelhalter, D. J., A. Thomas, N. G. Best, and W. R. Gilks, *BUGS 0.5: Baysian Inference using Gibbs Sampling*, MRC Biostatics Unit, Cambridge, UK, 1996.
[17] Taylor, G. C., *Loss Reserving: An Actuarial Perspective*, Norwell, MA: Kluwer, 2000.
[18] Taylor, G. C. and F. R. Ashe, "Second Moments of Estimates of Outstanding Claims," *Journal of Econometrics* 23, 1983, pp. 37–61.
[19] Verrall, R. J., "A Bayesian Generalised Linear Model for the Bornhuetter-Ferguson Method of Claims Reserving," *North American Actuarial Journal* 8:3, 2004, pp. 67–89.
[20] Verrall, R. J., "An Investigation Into Stochastic Claims Reserving Models and the Chain-Ladder Technique," *Insurance: Mathematics and Economics* 26, 2000, pp. 91–99.
[21] Verrall, R. J. and P. D. England, "Incorporating Expert Opinion into a Stochastic Model for the Chain-Ladder Technique," *Insurance: Mathematics and Economics* 37, 2005, pp. 355–370.

Appendix

The code for winBUGS is shown below for the models used in Section 6. This is available from the author on request and can be cut and pasted directly into winBUGS. Anything to the right of "#" is ignored, so the code can be changed by adding and removing this at the start of a line.

(i) This section contains the code for the Bornhuetter-Ferguson method in Section 5, which was used for the illustrations in Sections 6.1 and 6.3.

```
model
{
#Model for Data
        for( i in 1 : 45 ) {
        Z[i]<-Y[i]/1000
        pC[i]<-D[i]/1000
#Zeros trick
        zeros[i]<-0
        zeros[i]~dpois(phi[i])
        phi[i]<-(-pC[i]*log(1/(1+g[row[i]]))-Z[i]*log(g[row[i]]/(1+g[row[i]])))/scale
                }
#Cumulate down the columns:
        DD[3]<-DD[1]+Y[46]
        for( i in 1 : 2 ) {DD[4+i]<-DD[4+i-3]+Y[49+i-3]}
        for( i in 1 : 3 ) {DD[7+i]<-DD[7+i-4]+Y[52+i-4]}
        for( i in 1 : 4 ) {DD[11+i]<-DD[11+i-5]+Y[56+i-5]}
```

```
        for( i in 1 : 5 ) {DD[16+i]<-DD[16+i-6]+Y[61+i-6]}
        for( i in 1 : 6 ) {DD[22+i]<-DD[22+i-7]+Y[67+i-7]}
        for( i in 1 : 7 ) {DD[29+i]<-DD[29+i-8]+Y[74+i-8]}
        for( i in 1 : 8 ) {DD[37+i]<-DD[37+i-9]+Y[82+i-9]}

#Needed for the denominator in definition of gammas
        E[3]<-E[1]*gamma[1]
        for( i in 1 : 2 ) {E[4+i]<-E[4+i-3]*gamma[2]}
        for( i in 1 : 3 ) {E[7+i]<-E[7+i-4]*gamma[3]}
        for( i in 1 : 4 ) {E[11+i]<-E[11+i-5]*gamma[4]}
        for( i in 1 : 5 ) {E[16+i]<-E[16+i-6]*gamma[5]}
        for( i in 1 : 6 ) {E[22+i]<-E[22+i-7]*gamma[6]}
        for( i in 1 : 7 ) {E[29+i]<-E[29+i-8]*gamma[7]}
        for( i in 1 : 8 ) {E[37+i]<-E[37+i-9]*gamma[8]}

        EC[1]<-E[1]/1000
        EC[2]<-sum(E[2:3])/1000
        EC[3]<-sum(E[4:6])/1000
        EC[4]<-sum(E[7:10])/1000
        EC[5]<-sum(E[11:15])/1000
        EC[6]<-sum(E[16:21])/1000
        EC[7]<-sum(E[22:28])/1000
        EC[8]<-sum(E[29:36])/1000
        EC[9]<-sum(E[37:45])/1000

#Model for future observations
        for( i in 46 : 90 ) {
                a1[i]<-max(0.01,a[row[i]]*DD[i-45]/(1000*scale))
                b1[i]<-1/(gamma[row[i]]*1000*scale)
                Z[i]~dgamma(a1[i],b1[i])
                Y[i]<-Z[i]
                fit[i]<-Y[i]
                        }
scale<-52.8615
#Convert row parameters to gamma using (5.6)
        for (k in 1:9) {
                gamma[k]<-1+g[k]
                g[k]<-u[k]/EC[k]
                a[k]<-g[k]/gamma[k]
                        }
#Prior distributions for row parameters.
        for (k in 1:9) {
                        u[k]~dgamma(au[k],bu[k])
```

```
        au[k]<-bu[k]*(ultm[k+1]*(1-1/f[k]))
        bu[k]<-(ultm[k+1]*(1-1/f[k]))/pow(ultsd[k+1],2)
            }
#The prior distribution can be changed by changing the data input values for the
#vectors ultm and ultsd

#Row totals and overall reserve
    R[1]<-0
    R[2]<-fit[46]
    R[3]<-sum(fit[47:48])
    R[4]<-sum(fit[49:51])
    R[5]<-sum(fit[52:55])
    R[6]<-sum(fit[56:60])
    R[7]<-sum(fit[61:66])
    R[8]<-sum(fit[67:73])
    R[9]<-sum(fit[74:81])
    R[10]<-sum(fit[82:90])
    Total<-sum(R[2:10])
    }

#DATA
list(
row=c(1,1,1,1,1,1,1,1,1,
2,2,2,2,2,2,2,2,
3,3,3,3,3,3,3,4,4,
4,4,4,4,5,5,5,5,5,
6,6,6,6,7,7,7,8,
8,9,1,2,2,3,3,3,4,4,4,
4,5,5,5,5,5,6,6,6,6,6,6,
7,7,7,7,7,7,7,8,8,8,8,8,
8,8,8,9,9,9,9,9,9,9,9,
9),
Y=c(352118,884021,933894,1183289,445745,320996,527804,266172,425046,
290507,1001799,926219,1016654,750816,146923,495992,280405,
310608,1108250,776189,1562400,272482,352053,206286,
443160,693190,991983,769488,504851,470639,
396132,937085,847498,805037,705960,
440832,847631,1131398,1063269,
359480,1061648,1443370,
376686,986608,
344014,
NA,
NA,NA,
```

```
NA,NA,NA,
NA,NA,NA,NA,
NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,NA),

D=c(357848,766940,610542,482940,527326,574398,146342,139950,227229,
709966,1650961,1544436,1666229,973071,895394,674146,406122,
1000473,2652760,2470655,2682883,1723887,1042317,1170138,
1311081,3761010,3246844,4245283,1996369,1394370,
1754241,4454200,4238827,5014771,2501220,
2150373,5391285,5086325,5819808,
2591205,6238916,6217723,
2950685,7300564,
3327371,
NA,
NA,NA,
NA,NA,NA,
NA,NA,NA,NA,
NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,NA),

DD=c(67948,
652275,NA,
686527,NA,NA,
1376424,NA,NA,NA,
1865009,NA,NA,NA,NA,
3207180,NA,NA,NA,NA,NA,
6883077,NA,NA,NA,NA,NA,NA,
7661093,NA,NA,NA,NA,NA,NA,NA,
8287172,NA,NA,NA,NA,NA,NA,NA,NA),

E=c(67948,
652275,NA,
686527,NA,NA,
1376424,NA,NA,NA,
1865009,NA,NA,NA,NA,
3207180,NA,NA,NA,NA,NA,
```

```
6883077,NA,NA,NA,NA,NA,NA,
7661093,NA,NA,NA,NA,NA,NA,NA,
8287172,NA,NA,NA,NA,NA,NA,NA,NA),

f=c(1.017724725, 1.095636823, 1.154663551, 1.254275641, 1.384498969,
1.625196481, 2.368582213, 4.138701016, 14.44657687),
ultm=c(NA,5500, 5500, 5500, 5500, 5500, 6000, 6000, 6000, 6000),
ultsd=c(NA,10000,10000,10000,10000,10000,10000,10000,10000,10000))
```

These values for the ultsd will give the chain-ladder results. To obtain the Bornhuetter-Ferguson results, replace the last line with the following line:

```
ultsd=c(NA,1,1,1,1,1,1,1,1,1))
```

The other illustration in section 6.3 uses:

```
ultsd=c(NA,1000,1000,1000,1000,1000,1000,1000,1000,1000))

#INITIAL VALUES
list(u=c(5500, 5500, 5500, 5500, 5500, 6000, 6000, 6000, 6000),
Z=c(NA,NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,
NA,NA,NA,NA,
NA,NA,NA,
NA,NA,
NA,
0,
0,0,
0,0,0,
0,0,0,0,
0,0,0,0,0,
0,0,0,0,0,0,
0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0))
```

(ii) Code for the model in section 4, which was used for the illustrations in section 6.2.

```
model
{
#Model for data:
        for( i in 1 : 45 ) {
                Z[i]<-Y[i]/(scale*1000)
```

```
            pC[i]<-D[i]/(scale*1000)
            C[i]<-Z[i]+pC[i]
      zeros[i]<-0
            zeros[i]~dpois(phi[i])
            phi[i]<-(loggam(Z[i]+1)+loggam(pC[i])-loggam(C[i])-
pC[i]*log(p1[row[i],col[i]])-Z[i]*log(1-p1[row[i],col[i]]))
            }
DD[3]<-DD[2]+Y[47]
      for( i in 1 : 2 ) {DD[4+i]<-DD[4+i-1]+Y[49+i-1]}
      for( i in 1 : 3 ) {DD[7+i]<-DD[7+i-1]+Y[52+i-1]}
      for( i in 1 : 4 ) {DD[11+i]<-DD[11+i-1]+Y[56+i-1]}
      for( i in 1 : 5 ) {DD[16+i]<-DD[16+i-1]+Y[61+i-1]}
      for( i in 1 : 6 ) {DD[22+i]<-DD[22+i-1]+Y[67+i-1]}
      for( i in 1 : 7 ) {DD[29+i]<-DD[29+i-1]+Y[74+i-1]}
      for( i in 1 : 8 ) {DD[37+i]<-DD[37+i-1]+Y[82+i-1]}

#Model for future observations
      for( i in 46 : 90 ) {
            a1[i]<-max(0.01,(1-p1[row[i],col[i]])*DD[i-45]/(1000*scale))
                  b1[i]<-p1[row[i],col[i]]/(1000*scale)
                  Z[i]~dgamma(a1[i],b1[i])
                  Y[i]<-Z[i]
                        }
      scale<-52.8615

#Set up the parameters of the negative binomial model.
      for (k in 1:9) {
            p[k]<-1/lambda[k]
            lambda[k]<-exp(g[k])+1
            g[k]~dnorm(0.5,1.0E-6)
      }
#Choose one of the folllowing (1,2 or 3) and delete the "#" at the start of each line before running.

#1. Vague Priors: Chain-ladder model
#       for (j in 1:9) {
#       for (i in 1:10) {p1[i,j]<-p[j]}
#                       }

#2. Intervention in second development factor.
#       for (i in 1:10) {p1[i,1]<-p[1]}
#       for (i in 1:6) {p1[i,2]<-p[2]}
#       p1[7,2]<-p82
#       p1[8,2]<-p82
```

```
#       p1[9,2]<-p82
#       p1[10,2]<-p82
#       for (j in 3:9) {
#       for (i in 1:10) {p1[i,j]<-p[j]}
#                       }
#       lambda82<-g82+1
#       p82<-1/lambda82
#Use one of the following 2 lines:
#       g82~dgamma(0.005,0.01) #This is a prior with a large variance
#       g82~dgamma(25,50) #This is a prior with a small variance

#3. Using latest 3 years for estimation of development factors.
#       for (j in 1:6) {
#       for (i in 1:(7-j)) {p1[i,j]<-op[j]}
#       for (i in (8-j):10) {p1[i,j]<-p[j]}
#       }
#       for (j in 7:9) {
#       for (i in 1:10) {p1[i,j]<-p[j]}
#               }
#       for (k in 1:6) {
#               op[k]<-1/olambda[k]
#               olambda[k]<-exp(og[k])+1
#               og[k]~dnorm(0.5,1.0E-6)
#                       }

#Row totals and overall reserve
                R[1]<-0
                R[2]<-Y[46]
                R[3]<-sum(Y[47:48])
                R[4]<-sum(Y[49:51])
                R[5]<-sum(Y[52:55])
                R[6]<-sum(Y[56:60])
                R[7]<-sum(Y[61:66])
                R[8]<-sum(Y[67:73])
                R[9]<-sum(Y[74:81])
                R[10]<-sum(Y[82:90])
                Total<-sum(R[2:10])
        }
#DATA
list(
row=c(1,1,1,1,1,1,1,1,1,1,
2,2,2,2,2,2,2,2,2,
3,3,3,3,3,3,3,3,4,4,
```

```
4,4,4,4,5,5,5,5,5,
6,6,6,6,7,7,7,8,
8,9,2,3,3,4,4,
4,5,5,5,5,6,6,6,6,6,
7,7,7,7,7,7,8,8,8,8,
8,8,8,9,9,9,9,9,9,9,
9,10,10,10,10,10,10,10,10,10),
col=c(1,2,3,4,5,6,7,8,9,
1,2,3,4,5,6,7,8,
1,2,3,4,5,6,7,1,2,3,
4,5,6,1,2,3,4,5,1,
2,3,4,1,2,3,1,
2,1,9,8,9,7,8,9,
6,7,8,9,5,6,7,8,9,4,
5,6,7,8,9,3,4,5,6,7,
8,9,2,3,4,5,6,7,8,9,
1,2,3,4,5,6,7,8,9),
Y=c(
766940,610542,482940,527326,574398,146342,139950,227229,67948,
884021,933894,1183289,445745,320996,527804,266172,425046,
1001799,926219,1016654,750816,146923,495992,280405,
1108250,776189,1562400,272482,352053,206286,
693190,991983,769488,504851,470639,
937085,847498,805037,705960,
847631,1131398,1063269,
1061648,1443370,
986608,
NA,
NA,NA,
NA,NA,NA,
NA,NA,NA,NA,
NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,NA),
D=c(
357848,1124788,1735330,2218270,2745596,3319994,3466336,3606286,3833515,
352118,1236139,2170033,3353322,3799067,4120063,4647867,4914039,
290507,1292306,2218525,3235179,3985995,4132918,4628910,
310608,1418858,2195047,3757447,4029929,4381982,
443160,1136350,2128333,2897821,3402672,
396132,1333217,2180715,2985752,
```

```
440832,1288463,2419861,
359480,1421128,
376686,
NA,
NA,NA,
NA,NA,NA,
NA,NA,NA,NA,
NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,NA),
DD=c(5339085,
4909315,NA,
4588268,NA,NA,
3873311,NA,NA,NA,
3691712,NA,NA,NA,NA,
3483130,NA,NA,NA,NA,NA,
2864498,NA,NA,NA,NA,NA,NA,
1363294,NA,NA,NA,NA,NA,NA,NA,
344014,NA,NA,NA,NA,NA,NA,NA,NA))
```

#INITIAL VALUES
This is what is used for 1.

For 2, replace the first line by
list(g=c(0,0,0,0,0,0,0,0,0), g82=0.5,

For 3, replace the first line by
list(g=c(0,0,0,0,0,0,0,0,0), og=c(0,0,0,0,0,0),

```
list(g=c(0,0,0,0,0,0,0,0,0),
Z=c(NA,NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,NA,
NA,NA,NA,NA,NA,
NA,NA,NA,NA,
NA,NA,NA,
NA,NA,
NA,
0,
0,0,
```

0,0,0,
0,0,0,0,
0,0,0,0,0,
0,0,0,0,0,0,
0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,
0,0,0,0,0,0,0,0,0))