Loss distributions modeling for motor TPL insurance class using Gaussian Mixture Method and EM Algorithm

Sandra Teodorescu, Faculty of Economic Sciences, Ecological University of Bucharest, Romania
e-mail: cezarina_teodorescu@yahoo.com

Abstract

The motor insurance is an important branch of non-life insurance in many countries; in some of them, coming first in total premium income category (in Romania, for example). In this paper we present the Gaussian mixture method to model the loss distribution of data from motor compulsory third part liability insurance. The parameters of the mixture are estimated using the Expectation Maximization (EM) algorithm.

Keywords: loss distributions, expectation maximization (EM) algorithm, motor third part liability insurance.

1. Introduction

The motor compulsory third part liability insurance (MTPL insurance) is the sole Romanian obligatory insurance, at least until the mandatory home insurance comes into force, in the 2009 Spring.

During the previous year, the Romanian MTPL insurance owning a market share of 20,77% from the Romanian insurance industry and respectively 36,40% from the Romanian motor insurance market.

Every year, the insurance companies pay damages that are larger than the policy earnings. The estimation for the 2008 of the average damage rate showed an increase by 15-20% of this indicator. The damage rate has continuously grown due to the large number of accidents, the relatively small amount of insurance premiums on the market and due to the drivers' lack of discipline. Thus, the damage rate is around 65-70% for the MTPL insurance. Although the insurers have increased the MTPL tariffs every year, the policies are still an unprofitable line of business due to the high damages.

As specified in Klugman et al. [5], „*in the most general sense, all of actuarial science is about loss distributions because that is precisely what an insurance agreement is all about*". The policy holder is paid a random amount (the loss) at a random future time. Hence, a loss distribution is considered to be the probability distribution of either the loss, or the amount paid from a loss event. Evaluating the loss distribution for an homogeneous portfolio is of great importance for the insurance company, because this distribution is involved in developing probability distributions for the aggregate loss, and therefore in evaluating ruin probabilities, reserves, benefits etc., or in establishing the influence of different deductibles.

In this paper we will consider the finite mixture method to model the loss distribution of data from automobile liability insurance. The data were kindly provided by a Romanian insurance company and consists of all the liability claims settled for an entire portfolio.

In section 2 we present two examples, two different approaches to fit a loss distribution using a set of real data regarding the damages that were paid by a Romanian insurance company, for MTPL insurance.

In section 3 we present the the methods that we used: the Gaussian Mixture Method (GMM) used to approximate the density of the loss distribution, and the EM algorithm to compute the maximum likelihood estimators of the parameters of the GMM.

In section 4 we discuss the results.

2. Paper content

In this section we analyze two examples, two different approaches to fit loss distribution, using a set of real data regarding the damages that were paid by a Romanian insurance company, for MTPL insurance.

The data set consists of 1161 settled claims. The main empirical characteristics of this data set are:

Expected value=17,126,337.4
Standard deviation=24,267,282.3
Skewness=4.62
Standard Error Skewness=0.07
Kurtosis= 32.80
Standard Error Kurtosis= 0.14
Maximum value=310,000,000
Minimum value=9,000

In the following, we will consider two different approaches to fit a distribution: in the first approach we use the raw data, in the second approach we use the log data.

First approach: raw data

Using the algorithm described in section 3, we tried to fit the mixed density (2) to our data set for two different values of c: c=2 and c=15. The fitting results are presented in Figures 1 and 2, while Table 1 contains the mixtures parameters.

The histogram below shows data plotted on standard arithmetic scales.
The solid lines represent the probability densities estimated using the finite mixture method. Here the mixed densities are univariate normal distributions, i.e.

$$\hat{f}(x) = \sum_{i=1}^{c} w_i \phi\left(x, \mu_i, \sigma_i^2\right).$$

The number of densities, the mixing coefficients and the parameters of the densities are estimated using the EM adaptive algorithm. The red line correspond to c=2 densities and the green one to c=15 densities.

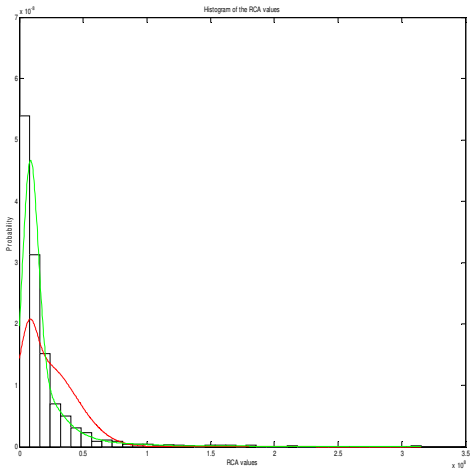

Fig 1. Histogram of the data set of settled claims (here we are using Scott's Rule for the bin widths).[1]

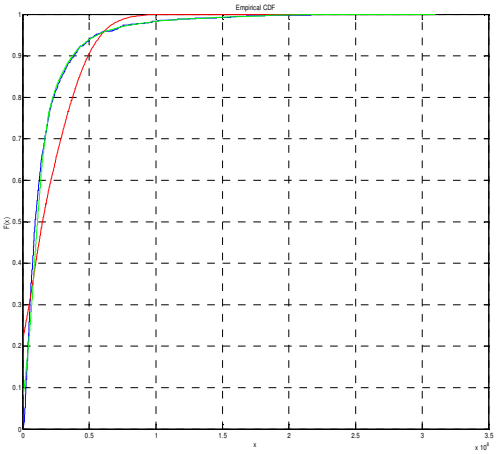

Fig 2. The empirical CDF (the staircase line in blue) versus the corresponding hypothesized CDF's (here the red line for c=2 and the green line for c=15).

It can be seen (Figure 1 and the result of the K-S test below) that if we use the raw data, the estimated cumulative density function and the corresponding hypothesized cumulative distribution function values computed using the GMM doesn't fit well. Though, if we must choose between the two theoretical models, the second one (c=15) clearly performs better (see Figures 1 and 2).

Table 1: The parameters of the mixtures from Fig 1

	w	μ	σ^2
c=2	0.86922	1.8548e+007	6.4899e+014
	0.13078	7.6754e+006	3.9614e+013
c=15	0.41895	8.7899e+006	4.2195e+013
	0.24905	7.9833e+006	3.8793e+013
	0.22368	1.9505e+007	1.6001e+014
	0.0028703	1.6429e+008	1.9308e+014
	0.046678	3.972e+007	1.5941e+014
	0.026936	5.6589e+007	1.9686e+014
	0.0098806	9.3195e+007	1.6907e+014
	0.0067904	7.4137e+007	1.5918e+014
	0.0027819	1.4544e+008	2.1019e+014
	0.003513	5.083e+006	7.2005e+013
	0.0031217	1.1004e+008	1.4911e+014
	0.0023252	1.268e+008	1.7793e+014
	0.00086133	3.1e+008	1.9843e+014
	0.00089951	2.1517e+008	2.017e+014
	0.0016588	1.7918e+008	1.9407e+014

The results of the Kolmogorov-Smirnov test (here $H_0 : F(x) = G(x)$; $H_1 : F(x) \neq G(x)$ where $F(x)$ is the estimated cumulative density function –cdf- and $G(x)$ the corresponding hypothesized cumulative distribution function values computed using the finite mixture method) with the significance level $\alpha = 0.05$ are given below.

	H_0	p-value	KSSTAT[2]	CV[3]
c=2	false	3.4568e-049	0.2191	0.039712
c=15	false	9.4979e-008	0.084911	0.039712

[1] *RCA values is the Romanian term for MTPL values.*

[2] *KSSTAT -the observed Kolmogorov-Smirnov statistic;*

[3] *CV - the cutoff value for determining if KSSTAT is significant.*

Second approach: log data

This time we used the same algorithm described in section 3, but we tried to fit the mixed density (2) to our log-data set, again for two different values of c: c=2 and c=12. The fitting results are presented in Figures 3 and 4, while Table 2 contains the mixtures parameters.

A logarithmic scale is a scale of measurement that uses the logarithm of a physical quantity instead of the quantity itself. Presentation of a data on a logarithmic scale can be helpful when the data covers a large range of values, just like in our case; the logarithm reduces this to a more manageable range. As it has been noticed, our dates make logarithmic scales especially appropriate.

The two histograms below demonstrate the difference between the two scales when plotting the same "claims" data, i.e. the liability claims settled. In terms of comparison, the histogram of the log data provides a more complete description of the data since it looks more compact by pooling the data, and allows a better fit of the model.

However, the logarithmic chart, using a logarithmic scale on the y axis, shows the percentage changes in the same rates.

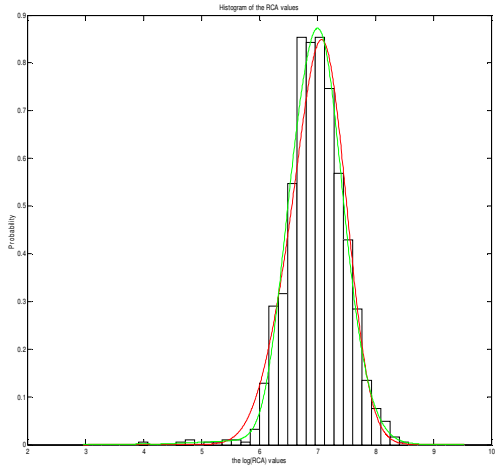

Fig 3 Histogram of the second log data set of settled claims (here we are using Scott's Rule for the bin widths).

The solid lines represent the probability densities estimated using the finite mixture method, here the mixed densities are univariate normal distributions, i.e.

$$\hat{f}(x) = \sum_{i=1}^{c} w_i \phi\left(x, \mu_i, \sigma_i^2\right).$$

The number of densities, the mixing coefficients and the parameters of the densities are estimated using the EM adaptive algorithm. The red line correspond to c=2 densities and the green one to c=12 densities.

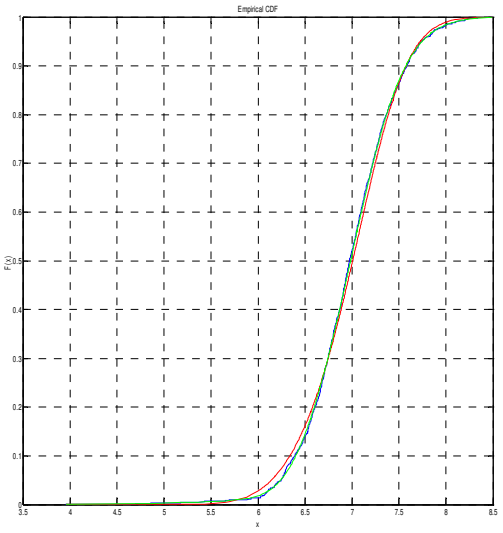

Fig 4. The empirical CDF (the staircase line in blue) versus the corresponding hypothesized CDF's (here the red line for c=2 and the green line for c=12).

Table 2: The parameters of the mixtures from Fig 3

	w	μ	σ^2
c=2	0.75048	7.1363	0.15337
	0.24952	6.5099	0.16683
c=12	0.48756000	7.1880	0.107080
	0.20362000	6.5503	0.066176
	0.21833000	6.8967	0.073772
	0.02686500	6.2430	0.062181
	0.01609100	7.9387	0.064838
	0.03285600	7.6770	0.062694
	0.00461110	5.8356	0.092219
	0.00356900	5.3295	0.081150
	0.00211490	8.1409	0.070830
	0.00131410	8.4006	0.069821
	0.00222120	4.7950	0.074949
	0.00086133	3.9542	0.074958

The results of the Kolmogorov-Smirnov test with the significance level $\alpha = 0.05$ are given below.

	H_0	p-value	KSSTAT	CV
c=2	true	0.1753	0.032258	0.039712
c=12	true	0.99227	0.012623	0.039712

We noticed that this time both models fit the data, but the second one (c=12) fits better.

3. Theoretical Background - Research methods

The Gaussian Mixture Method

The finite mixture method assumes the density *f(x)* can be modeled as the sum of *c* weighted densities, with $c \square n$. The most general case for the univariate finite mixture is

$$f(x) = \sum_{i=1}^{c} p_i g(x;\theta_i) \qquad (1)$$

where p_i represents the *weight* or *mixing coefficient* for the *i*-th term, and $g(x;\theta_i)$ denotes a probability density, with parameters represented by the vector θ_i. To make sure that this is a *bona fide* density, we must impose the condition that $p_1 + ... + p_c = 1$ and $p_i > 0$. To evaluate $f(x)$, we take our point *x*, find the value of the component densities $g(x;\theta_i)$ at that point, and take the weighted sum of these values.

The component densities of the finite mixture can be any probability density function, continuous or discrete. In this paper, we confine our attention to the continuous case and use the *normal density* for the component function. Therefore, the estimate of a finite mixture would be written as

$$\hat{f}_{FM}(x) = \sum_{i=1}^{c} \hat{p}_i \varphi(x;\hat{\mu}_i, \hat{\sigma}_i^2) \qquad (2)$$

where $\varphi(x;\hat{\mu}_i, \hat{\sigma}_i^2)$ denotes the normal probability density function with mean $\hat{\mu}_i$ and variance $\hat{\sigma}_i^2$. In this case, we have to estimate *c*-1 independent mixing coefficients, as well as the *c* means and *c* variances using the data. Note that to evaluate the density estimate at a point *x*, we only need to retain these 3*c*-1 parameters. With finite mixtures much of the computational burden is shifted to the estimation part of the problem.

The EM Training Algorithm

The method for determining the parameters of a finite mixture of normal densities (i.e. a Gaussian mixture method – GMM) from a data set is based on maximizing the data likelihood. It is convenient to recast the problem in the equivalent form of minimizing the negative log likelihood of the data set

$$E = -L = -\sum_{j=1}^{n} \log f(X_j)$$

which is treated as an error function. There are two practical difficulties with this minimization problem. Firstly, the global minimum of E is $-\infty$. This is achieved when one of the Gaussian components collapses onto a data point so that $\mu_i = x$ and the corresponding variance tends to 0. To avoid this problem, the variance is checked at each iteration, and dangerously small values are replaced by larger ones. Secondly, there are often a large number of local minima which correspond to poor models of the true density function. A solution for this is to train models from many different starting points and to take care over the initialization of the models.

Because the likelihood is a differentiable function of the parameters, it is possible to use a general purpose non-linear optimizer to find the minima of *E*. However, there are some advantages to using a specialized method, known as the *expectation-maximization*, or EM, algorithm (Dempster et al. [9]). This algorithm is simple to implement and understand, avoids the calculation and storage of derivatives, is usually faster to converge than general purpose algorithms, and can also be extended to deal with data sets where some points have missing values. The ideas behind the algorithm have also been applied to many other probabilistic models, including hidden Markov models and Kalman filters. With the use of variational methods, the EM algorithm has recently been extended to provide upper bounds on the error function *E* for classes of so-called *probabilistic graphical models* where the exact calculation of *E* is computationally intractable.

The EM algorithm iteratively modifies the GMM parameters to decrease *E*. It is guaranteed to reduce *E* at each step until a local minimum is found. It is helpful to suppose that the data set was sampled from an (unknown) mixture model. If we knew which component each data point X_j had been sampled from, then it would be straightforward to estimate the model parameters. Let I_i denote the indices of the data points sampled from component *i*, and *n* the total number of data points. Then the prior p_i would be given by

$$p_i = \frac{|I_i|}{n}$$

the mean μ_i by

$$\mu_i = \frac{1}{|I_i|}\sum_{j\in I_i} X_j$$

and the covariance by a similar formula that depends on the form of the covariance matrix; for example, for spherical covariance

$$\sigma_i^2 = \frac{1}{|I_i|}\sum_{j\in I_i} \left\| X_j - \mu_i \right\|^2 .$$

Of course, we don't know which component generated each data point, so instead we consider a hypothetical *complete* data set in which each data point is labeled with the component that generated it. So, for each data point X_j, there is a corresponding random variable z_j, which is an integer in the range $1,\ldots,c$. We write y_j for the complete data point (X_j, z_j) and w for the parameters in the mixture model. The EM algorithm generates a sequence of estimates $w^{(m)}$ starting from the initial parameter set $w^{(0)}$.

First we write down the likelihood of a complete data point if $z = i$:

$$\begin{aligned} p\left((x, z=i)\middle|w\right) &= p\left(x\middle|z=i,w\right) p\left(z=i\middle|w\right) \\ &= p\left(x\middle|\theta_i\right) p\left(z=i\middle|w\right) \end{aligned}$$

where θ_i are the density function parameters (mean and variance) for *i* component. The likelihood of *x* can be obtained by marginalizing over *z* which, since it is a discrete variable, is simply a matter of summing the above formula over all its possible values:

$$p\left(x\middle|w\right) = \sum_{j=1}^{c} p\left(z=i\middle|w\right) p\left(x\middle|\theta_i\right)$$

Comparing this with (1), we see that the probabilities $p\left(z=i\middle|w\right)$ are playing the same role as the mixing coefficients.

Given a set of parameters $w^{(m)}$ we would like to use class labels z_i and the above formulas of the p_i, μ_i and σ_i^2 to estimate the next set of parameters $w^{(m+1)}$. As we don't know the class labels, but do know their probability distribution, what we can do is to use the expected values of the class labels given the current parameters. We form the function $Q\left(w\middle|w^{(m)}\right)$ as follows:

$$\begin{aligned} Q\left(w\middle|w^{(m)}\right) &= E\left(\log p\left(y\middle|w\right)\right) p\left(z_j\middle|X_j, w^{(m)}\right) \\ &= \sum_{i=1}^{c}\sum_{j=1}^{n}\left[\log p\left(X_j, z_j\middle|w\right)\right] p\left(z_j\middle|X_j, w^{(m)}\right) \\ &= \sum_{i=1}^{c}\sum_{j=1}^{n}\left[\log p_i + \log p\left(X_j\middle|\theta_i\right)\right] p^{(m)}\left(i\middle|X_j\right), \end{aligned}$$

where

$$p^{(m)}\left(i\middle|X_j\right) := p\left(z_j = i\middle|X_j, w^{(m)}\right) = \frac{p_i^{(m)} p\left(X_j\middle|\theta_i^{(m)}\right)}{\sum_{i=1}^{c} p_i^{(m)} p\left(X_j\middle|\theta_i^{(m)}\right)}$$

is the expected posterior distribution of the class labels given the observed data. Note that $Q\left(w\middle|w^{(m)}\right)$ is a function of the parameters p_i and θ_i while $p_i^{(m)}$ and $\theta_i^{(m)}$ are fixed values. The calculation of *Q* is the E-step of the algorithm. To compute the new set of parameter values $w^{(m+1)}$, we optimize $Q\left(w\middle|w^{(m)}\right)$, i.e.

$$w^{(m+1)} = \arg\min_{w} Q\left(w\middle|w^{(m)}\right).$$

This is the M-step of the algorithm.

To use the EM algorithm, we must have a value for the number of terms *c* in the mixture. This is usually obtained using prior knowledge of the application (the analyst expects a certain number of groups), using graphical exploratory data analysis (looking for clusters or other group structure) or using some other method of estimating the number of terms. In our case we consider *c=2.*

Besides the number of terms, we must also have an initial guess for the value of the component parameters. Once we have an initial estimate, we update the parameter estimates using the data and the equations given below. These are called the iterative EM update equations, and we provide the univariate case. The multivariate case follows easily. The first step is to determine the posterior probabilities given by

$$\hat{\tau}_{ij} = \frac{\hat{p}_i \varphi\left(\mathbf{x}_j; \hat{\mu}_i, \sigma_i^2\right)}{\hat{f}\left(\mathbf{x}_j\right)}; \quad i = 1,...,c; j = 1,...,n \qquad (3)$$

where $\hat{\tau}_{ij}$ represents the estimated posterior probability that point $\mathbf{x}_j$ belongs to the *i*-th term, $\varphi\left(\mathbf{x}_j; \hat{\mu}_i, \sigma_i^2\right)$ is the normal density for the *i*-th term evaluated at $\mathbf{x}_j$, and

$$\hat{f}\left(\mathbf{x}_j\right) = \sum_{k=1}^{c} \hat{p}_k \varphi\left(\mathbf{x}_j; \hat{\mu}_k, \sigma_k^2\right) \qquad (4)$$

is the finite mixture estimate at point $\mathbf{x}_j$.

The posterior probability tells us the likelihood that a point belongs to each of the separate component densities. We can use this estimated posterior probability to obtain a weighted update of the parameters for each component. This yields the iterative EM update equations for the mixing coefficients, the means and the covariance matrices. These are

$$\hat{p}_i = \frac{1}{n} \sum_{j=1}^{n} \hat{\tau}_{ij}, \qquad (5)$$

$$\hat{\mu}_i = \frac{1}{n} \sum_{j=1}^{n} \frac{\hat{\tau}_{ij} \mathbf{x}_j}{\hat{p}_i}, \qquad (6)$$

$$\hat{\sigma}_i^2 = \frac{1}{n} \sum_{j=1}^{n} \frac{\hat{\tau}_{ij} \left(x_j - \hat{\mu}_i\right)^2}{\hat{p}_i} \qquad (7)$$

The steps for the EM algorithm to estimate the parameters for a finite mixture with multivariate normal components are given here (see Enachescu, [3]).

EM Algorithm for finite mixtures :

Step 1 Determine the number of terms or component densities *c* in the mixture.

Step 2 Determine an initial guess at the component parameters. These are the mixing coefficients, means and covariance matrices for each normal density.

Step 3 For each data point $\mathbf{x}_j$, calculate the posterior probability using Equation 3.

Step 4 Update the mixing coefficients, the means and the covariance matrices for the individual components using Equations 5 through 7.

Step 5 Repeat steps 3 through 4 until the estimates converge.

Typically, step 5 is implemented by continuing the iteration until the changes in the estimates at each iteration are less than some pre-set tolerance. Note that with the iterative EM algorithm, we need to use the entire data set to simultaneously update the parameter estimates. This imposes a high computational load when dealing with massive data sets.

4. Conclusions

From Figure 1, we notice that the raw data set histogram corresponds to a strongly asymmetric density with right heavy tail. Also, based on the fact that the Kolmogorov-Smirnov test rejected the density (2) assumption for raw data, we considered that applying the logarithmic function to the initial data (possible since all data are positive) we could pool the data and obtain a better fit of the model. Our procedure proved to be correct, since the Kolmogorov-Smirnov test accepted the density (2) assumption for the log-data. The best fit is for *c*=12, as expected (usually, more parameters means a better fit).

In conclusion, our log-data can be modelled by a Gaussian mixture model, which means that the raw data could be modelled by a product of lognormals model.

5. References

[1] Bohning, D., Dietz, E., Kuhnert R., and Schon, D. "Mixture models for capture –recapture count data" *Statistical Methods & Applications* 14, 2005, p.29-43 .

[2] Dempster, A.P., Laird, M., and Rubin, D.B. „Maximum likelihood from incomplete data via the EM algorithm*" Journal of the Royal Statistical Society*, B 39 (1) ,1977, p. 1-38.

[3] Enăchescu, D. „*Unsupervised Statistical Learning and data mining"* CLEUP (Cooperativa Libraria Editrice Universita di Padova) , 2004.

[4] Kaas, R., Goovaerts, M., Denuit, M., and Dhaene, J. „*Modern Actuarial Risk Theory",* Kluwer Academic Publishers, Boston, 2001.

[5] Klugman, S.A., Panjer, H.H., and Willmot, G., „*Loss Models. From data to decisions*." Wiley-Interscience, New York, 1998.

[6] Ross, S. „*Simulation"*, Second Edition, N.Y, Academic Press, 1997.

[7] Saporta, G., Preda, V. „A general family of probability distributions", *Publications de l'Institute*

de Statistique de l'Universite de Paris, XXXIX(2), 1995, p. 71-93.

[8] Teodorescu, S., and Vernic, R. „The EM algorithm for loss distributions modelling", *Preprint Series in Computer Science and Applied Mathematics*, „Ovidius" University from Constanta , Romania, 2005.