Loss Forecasting Model

Simulation Modeling for Financial Stress Testing
using Machine Learning and Copulas

Alvin Chung

September 8, 2018

Contents

Abstract

The ability to accurately estimate the VaR (Value at Risk) of a financial portfolio has always been a difficult challenge. Traditionally, techniques such as linear and generalised linear models, Poisson process and Markov process have been used for actuarial modeling in estimating claim size models, claim frequency models, loss reserve forecasting, pure premium calculation. These methodologies are strong in the fact that they provide significant interpretablity. However, they suffer from numerous weaknesses. Most notably, they are often poor estimates due to the fact that they are extremely sparse, and are prone towards underfitting to meet the requirements for Monte Carlo Simulations.

In this paper, we present an alternative approach towards conducting stress testing of Mortgage Backed Securities through the use of Machine Learning in forecasting expected changes property prices and using those estimations with a copula in deriving the value at risk (VaR) for a financial portfolio of mortgage backed securities. The resulting methodology **should** be capable of correctly estimating the casual relationship among various Mortgage Backed Securities that may effect the portfolio and thus, simulating stress testing scenarios with a higher accuracy and lower computational complexity than traditional Monte Carlo Simulations

0.1 Introductions

- ARIMA(1,0,1)
- Linear Regression
- Support Vector Machine (Classification)
- GANS (Maybe)

0.2 Theory and Definitions

0.2.1 Linear Regression

Regression is one of the most widely used of all statistical methods. For univariate regression, the available data are one response variable and p predictor variables, all measured on each of n observations. We let Y denote the response variable and $X_1, ..., X_p$ be the predictor or explanatory variables. Also, Y_i and $X_i, 1, ..., X_{i,p}$ are the values of these variables for the ith observation. The goals of regression modeling include the investigation of how Y is related to $X_1, ..., X_p$, estimation of the conditional expectation of Y given $X_1, ..., X_p$, and prediction of future Y values when the corresponding values of $X_1, ..., X_p$ are already available. These goals are closely connected. The multiple linear regression model relating Y to the predictor or regression variables is

$$\mathbf{Y}_i = \beta_0 + \beta_1 \mathbf{X}_{i,1} + \cdots + \beta_p \mathbf{X}_{i,p} + \epsilon_i,$$

where ϵ_i is called the noise, disturbances, or errors. The adjective "multiple"

refers to the predictor variables. Multivariate regression, which has more than one response variable. The ϵ_i are often called "errors" because they are the prediction errors when Y_i is predicted by $\beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}$. It is assumed that,

$$E(\epsilon_i | X_{i,1}, ..., X_{i,p}) = 0,$$

The parameter β_0 is the intercept. The regression coefficients $\beta_1, ..., \beta_p$ are the slopes. More precisely, β_j is the partial derivative of the expected response with respect to the jth predictor:

Therefore, β_j is the change in the expected value of Y_i when $X_{i,j}$ changes one unit. It is assumed that the noise is *i.i.d.* white so that:

$$\epsilon_1, ..., \epsilon_n \text{are i.i.d. with mean 0 and variance } \sigma_\epsilon^2.$$

Often the ϵ_is are assumed to be normally distributed, which with implies Gaussian white noise. For convenience, the assumptions of the linear regression model are summarized:

1. linearity of the conditional expectation: $E(Y_i | X_{i,1}, ..., X_{i,p}) = \beta_0 + \beta_1 X_{i,1} + \cdots + \beta_p X_{i,p}$

2. independent noise: $\epsilon_1, ..., \epsilon_n$ are independent

3. constant variance: $Var(\epsilon_i) = \sigma\epsilon_2$ for all i

4. Gaussian noise: ϵ_i is normally distributed for all i.

The amount of variation in Y that cannot be predicted by a linear function of

$X_1, ..., X_p$ is measured by the residual error sum of squares, which we will use the mean squared error:

$$\sum_{i=1}^{n} \frac{1}{n}(Y_i - \hat{Y}_i)^2$$

This is the error term we are trying to minimize when fitting our model.

0.2.2 The Gauss–Markov Theorem

One of the most famous results in statistics asserts that the least squares estimates of the parameters β have the smallest variance among all linear unbiased estimates. We will make this precise here, and also make clear that the restriction to unbiased estimates is not necessarily a wise one. This observation will lead us to consider biased estimates such as ridge regression later in the chapter. We focus on estimation of any linear combination of the parameters $\theta = a^T\beta$ for example, predictions $f(x_0) = x_0^T\beta$ are of this form. The least squares estimate of aT β is

$$\hat{\theta} = a^T\hat{\beta} = a^T(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T y$$

Considering X to be fixed, this is a linear function c_0^T y of the response vector y. If we assume that the linear model is correct, a^T $\hat{\beta}$ is unbiased since

$$E(a^T\hat{\beta}) = E(a^T(X^TX)^{-1}X^Ty)$$

$$= a^T(X^TX)^{-1}XTX\beta$$

$$= a^T \beta.$$

The Gauss–Markov theorem states that if we have any other linear estimator $\theta = c^T y$ that is unbiased for $a^T \beta$, that is, $E(c^T y) = aT\beta$, then

$$Var(a^T \hat{\beta}) \leq Var(c^T y)$$

Thus, a linear regression model in which the errors have expectation zero, are uncorrelated and have equal variances, the best linear unbiased estimator (BLUE) of the coefficients is given by the ordinary least squares (OLS) estimator, provided it exists.

0.2.3 Subset Selection

There are two reasons why we are often not satisfied with the least squares estimates

• The first is prediction accuracy: the least squares estimates often have low bias but large variance. Prediction accuracy can sometimes be improved by shrinking or setting some coefficients to zero. By doing so we sacrifice a little bit of bias to reduce the variance of the predicted values, and hence may improve the overall prediction accuracy.

• The second reason is interpretation. With a large number of predictors, we often would like to determine a smaller subset that exhibit the strongest effects. In order to get the "big picture," we are willing to sacrifice some of the small details.

In this section we describe a number of approaches to variable subset selection with linear regression. In later sections we discuss shrinkage and hybrid approaches for controlling variance, as well as other dimension-reduction strategies. These all fall under the general heading model selection. Model selection is not restricted to linear models.

With subset selection we retain only a subset of the variables, and eliminate the rest from the model. Least squares regression is used to estimate the coefficients of the inputs that are retained. There are a number of different strategies for choosing the subset

0.2.4 Hybrid-Stepwise Selection

For this paper, we will use a Hybrid stepwise-selection strategies that consider both forward and backward moves at each step, and select the "best" of the two. In the R package the step function uses the AIC criterion for weighing the choices, which takes proper account of the number of parameters fit; at each step an add or drop will be performed that minimizes the AIC score.

0.2.5 Shrinkage Method

Under standard assumptions, the coefficients produced by ordinary least squares regression are unbiased and, of all unbiased linear techniques, this model also has the lowest variance. However, given that the MSE is a combination of variance and bias, it is very possible to produce models with smaller MSEs by allowing the parameter estimates to be biased. It is common that a small increase in bias can produce a substantial drop in the variance and thus a smaller MSE than ordinary least squares regression coefficients. One consequence of large correlations between the predictor vari- ances is that the variance can

become very large. Combatting collinearity by using biased models may result in regression models where the overall MSE is competitive.

One method of creating biased regression models is to add a penalty to the sum of the squared errors. Recall that original least squares regression found parameter estimates to minimize the sum of the squared errors:

$$SSE = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

When the model over-fits the data, or when there are issues with collinearity, the linear regression parameter estimates may become inflated. As such, we may want to control the magnitude of these estimates to reduce the SSE. Controlling (or regularizing) the parameter estimates can be accomplished by adding a penalty to the SSE if the estimates become large. Ridge regression (Hoerl 1970) adds a penalty on the sum of the squared regression parameters:

Ridge Regression

$$SSE_{L2} = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{P} \beta_j^2$$

The "L2" signifies that a second-order penalty (i.e., the square) is being used on the parameter estimates. The effect of this penalty is that the parameter estimates are only allowed to become large if there is a proportional reduction in SSE. In effect, this method shrinks the estimates towards 0 as the λ penalty becomes large (these techniques are sometimes called "shrinkage methods"). By adding the penalty, we are making a trade-off between the model variance and bias. By sacrificing some bias, we can often reduce the variance enough to make the overall MSE lower than unbiased models.

0.2.6 Support Vector Machine

Support vector machines are a class of statistical models first developed in the mid-1960's by Vladimir Vapnik. Suppose we have a two-class problem and we code the class #1 samples with the value of 1 and the class #2 samples with -1. Also, let the vector x_i contain the predictor data for the training set samples. The maximum margin classifier creates a decision value $D(x)$ that classifies samples such that if $D(x) > 0$, we would predict a sample to be class #, otherwise class #. For an unknown sample u, the decision equation can be written in a similar form as a linear discriminant function that is parametrized in terms of an intercept and slope as,

$$D(u) = \beta_o + \beta u = \beta_0 + \sum \beta_j u_j$$

Notice that this equation works from the viewpoint of the predictors. This equation can be transformed so that the maximum margin classifier can be written in terms of each data point in the sample. This changes the equation to,

$D(u) = \beta_0 + \sum \beta_j u_j$

$= \beta_0 + \sum y_i a_i x_i u$

where u_i is for the unknown sample, and x_i contain the predictor data for a training set sample, and $D(u)$ is the decision value for the classification of the unknown sample, where $a_i \geq 0$.

It turns out that, in the complete separable case, the a parameters are exactly zero for all samples that are not on the margin. Conversely, the set of nonzero a values are the points that fall on the boundary of the margin. Because of this,

the predictor equation is a function of only a subset of the training set points and these are referred to as the *support vectors*. Interestingly, the prediction function is only a function of the training set samples that are closest to the boundary and are predicted with the least amount of certainty.

Since the prediction equation is *supported* solely by these data points, the maximum margin classifier is usually called the *support vector machine*. On first examination, it may appear someone arcane. However, it can shed some light on how we support vector machines classify new samples.

Consider the following figure; Where a new sample, shown as a solid grey circle, is predicted by the model. The distance between each of the support vectors and the new sample are as grey dotted lines.

$$PictureMissing$$

For these data, there are three support vectors, and therefore contain the only information necessary for classifying the new sample. The meat of the equation for SVM is the summation of the product of the sign of the class, the model parameter, and the dot product between the new sample and the support vector predictor values. The following table shows the components of this sum, broken down for each of these three support vectors.

$$MissingPicture$$

The dot product, $x_i u$ can be written as a product of the distance of x_i from the origin, the distance of u from the origin, and the cosine of the angle between x_i and u.

$$x \cdot u = ||x|| \cdot ||u|| \cdot \cos(\theta)$$

Based on the parameter estimates a_i, the first support vector has the largest single effect on the prediction equation (all other things being equal) and it has a negative slope. For out new sample, the dot product is negative, so the total contribution of the point is positive and pushes the prediction towards the first class (.eg., a positive value of the decision function $D(u)$). The remaining two support vectors have positive dot products and an overall product that increases the decision function for this new sample. For this model, the intercept is -4.372; $D(u)$ for the new sample is therefore 0.583. Since this value is greater than zero, the new sample has the highest association with the first class.

What happens when the classes are not completely separable? there is a developed extension to the early maximum margin classifier to accommodate this situation. Their formulation puts a cost on the sum of the training set points that are on the boundary, or on the wrong side of the boundary. When determining the estimates of the a values, the margin is penalized when data points are on the wrong side of the class boundary or inside the margin. The cost value would be a tuning parameter for the model and is the primary mechanism to control the complexity of the boundary. For example, as the cost of errors increases, the classification boundary will shift and contort itself so that it correctly classifies as many of the training set points as possible.

Non-Linear Support Vector Machines

To extend the Support Vector Machine towards a non-linear decision boundary, we substitute the kernel function instead of a simple linear cross product.

$$D(u) = \beta_0 + \sum y_i a_i x_i u$$

$$= \beta_0 + \sum y_i a_i K(x_i u)$$

where $K(.,.)$ is a *kernel function* of the two vectors. For the linear case, the kernel function is the same inner product $x_i u$/ However, just as in regression SVMs, other non-linear transformations can be applied, including:

$$\text{polynomial} = (\text{scale}(xu) + 1)^{\text{degree}}$$

$$\text{Radial Basis Function} = exp(-\sigma||x - u||^2)$$

$$\text{hyperbolic tangent } = tanh(scale(xu) + 1)$$

Note that, due to the dot product, the predictor data should be centred and scaled prior to fitting so that attributes who's values are large in magnitude do not dominate the calculations.

The *kernel trick* allows the SVM model to produce extremely flexible decision boundaries. The choice of the kernel function parameters and the cost value control the complexity and should be tuned appropriately so that the model does not overfit the training data.

0.2.7 Copulas

Copulas are tools for modelling dependence of several random variables. The term *copula* was first used in the work of Sklar (1959) and is derived from the latin word *copulare*, to connect or to join. The main purpose of copulas is to describe the interrelation of several random variables.

0.2.8 Copulas Definition and Example

Let us start with an explanatory example: Consider two real-valued random variables X_1 and X_2 which shall give us two numbers out of $\{1, 2, ..., 6\}$. These numbers are the outcome of a more or less simple experiment or procedure. Assume that X_1 is communicated to us and we may enter a bet on X_2. The question is, how much information can be gained from the observation of X_1, or formulated in a different way, what is the interrelation or dependence of these two random variables (rvs).

The answer is easily found if the procedure is just that a dice is thrown and the outcome of the first throw is X_1 and the one of the second is X_2. In this case the variables are independent: the knowledge of X_1 gives us no information about X_2. The contrary example is when both numbers are equal, such that with X_1 we have full information on X_2.

A quite different answer will be given X_1 is always the number of the smaller throw and X_2 the larger one. Then we have a strict monotonic relation between these two, namely $X_1 \leq X_2$. In this case where $X_1 = 6$ we also know X_2. If $X_1 = 5$, we would guess that X_2 is either 5 or 6, both with a chance of 50%, and so on.

For a deeper analysis we will need some tools which help us to describe the

possible dependency of these two rvs. Observe that each rv is fully described by the cumulative distribution function (cdf) $F_i(X) := P(X_i \leq x)$ (the so-called *marginals*). In the case of throwing the dice twice we would typically have $F_1 = F_2 =: F$. However, and this is important to note, the cdfs give us no information about the joint behaviour. If we have independence as in the first example, the joint distribution function is simply the product of the marginals,

$$P(X_1 \leq x_1, X_2 \leq x_2) = F(X_1) \cdot F(X_2).$$

Hence, to obtain a full description of X_1 and X_2 together we used *two ingredients*: the marginals and the type of interrelation, in this case independence. The question is if this kind of separation between marginals and dependence can also be realized in a more general framework. Luckily the answer is yes, and the right concept for this is copulas.

To get this feeling about this, consider the third case in the above example, where $X_1(X_2)$ is the minimum (maximum) of the thrown dice, respectively. This is not difficult to deduce the joint distribution function,

$$P(X_1 \leq x_1, X_2 \leq x_2) = 2F(min\{x_1, x_2\})F(x_2) - F(min\{x_1, x_2\})^2.$$

Now, if the dice was numbered $11, 12, ..., 16$ instead of $1, 2, ..., 6$ the dependence structure obviously wouldn't change, but the joint distribution function would be totally different. This is due to the marginal distribution and dependence structure. The goal is to transform the rvs X_i into uniformly distributed rvs U_i. Which therefore, a rvs X with cdf F can always be represented as $X = F^{\leftarrow}(U)$, where $F^{\leftarrow}$ denotes the generalised inverse of F as defined below. Therefore the join distribution function can be restated, using two independent and standard uniformly distributed rvs U_1 and U_2, as

$$P(F_1^{\leftarrow}(U_1) \leq x_1, F_2^{\leftarrow}(U_2) \leq x_2) = P(U_1 \leq F_1(X_1), U_2 \leq F_2(x_2))$$

for example, comparing the expression for the independent case and the above representations of the joint distribution function we realise that the dependence structure itself - stripped from the marginals - may be expressed setting $u_i = F_i(x_i) = F(x_i)$ via

$$C(u_1, u_2) = u_1 \cdot u_2$$

and this function is a copula. The function describes the dependence structure seperated from the marginals. The intuition behind this is that the marginal distributions are transformed to uniform ones which are used as the reference case. The copula then expresses the dependence structure according to this reference case.

0.2.9 Sklar's theorem

Given the above result on quantile transformations, it is not surprising that every distribution function on $\Re^d$ inherently embodies a copula function. On the other side, if we choose a copula and some marginal distributions and entangle them in the right way, we will end up with a proper multivariate distribution function. This is due to the following theorem,

Theorem 2.3. Sklar (1959) Consider a d-dimensional *cdf* F with marginals $F_1, ..., F_d$. There exists a copula C, such that,

$$F(x_1, ..., x_d) = C(F_1(x_1), ..., F_d(x_d))$$

For all x_i in $[-\infty, \infty]$, $i = 1, ..., d$. If F_i is continuous for all $i = 1, ..., d$ then C is unique; otherwise C is uniquely determined only on $RanF_1 \times \cdots \times RanF_d$, where $RanF_i$ denotes the range of the cdf F_i. In the explanatory example with dice, the range was $\{\frac{1}{6}, \frac{2}{6}, ..., \frac{6}{6}\}^2$, while for a continuous rv this is always $[0, 1]$.

0.3 Method

0.3.1 Data Understanding

	state	address	date	property_prices
0	New South Wales	24/95 Euston Rd	2018-08-04	802000.0
1	New South Wales	608/222 Botany Rd	2018-08-04	525000.0
2	New South Wales	4/100 Buckland St	2018-08-04	1760000.0
3	New South Wales	11 Coachwood Cr	2018-08-04	1130000.0

	area	bathrooms	bed	bor_one_AQF_levels	bor_one_age	bor_one_credit_score
0	338	1	4	5	50	1022
1	190	1	2	6	34	978
2	196	2	1	7	44	1002
3	196	2	4	5	44	1065
4	284	1	4	8	42	1070

	bor_one_gender	bor_one_occuptation	bor_two_AQF_levels	bor_two_age
0	F	Press photographer	6	43
1	M	Neurosurgeon	8	48
2	M	Clinical embryologist	9	49
3	F	Psychotherapist, child	5	50
4	F	Pharmacist, community	8	33

	bor_two_credit_score	bor_two_gender	bor_two_occuptation	car_spaces	cpi
0	945	F	Systems developer	2	2.25
1	1013	F	Media buyer	2	2.00
2	1043	M	Plant breeder/geneticist	1	2.25
3	1033	M	Haematologist	1	1.70
4	919	F	Product manager	1	1.50

	gbp	hpi	interest	mortgage_type	property_type	result
0	2.00	1.50	2.25	Variable	Unit	Withdrawn
1	1.00	1.00	1.70	Variable	Unit	Withdrawn
2	2.00	2.50	0.75	Fixed	House	Passed In - No Bid
3	0.50	2.50	1.70	Fixed	House	Sold At Auction
4	0.75	0.75	2.50	Variable	Unit	Passed In - No Bid

0.3.2 Linear Regression

0.3.3 Support Vector Machine

0.3.4 Copulas

0.4 Result

0.5 Conclusion