Distance to default: Implementation in R

Ajay Shah
National Institute of Public Finance and Policy, Delhi

Manish Singh
Indira Gandhi Institute of Development Research, Mumbai

Nidhi Aggarwal
Indira Gandhi Institute of Development Research, Mumbai

Abstract

This paper describes the implementation of the function `dtd` in the package **ifrogs**. The function implements the Merton Model (1974) to derive the measure 'Distance to default' which can be used to assess the credit risk of a firm. The measure indicates how far is the firm from the default point.

Keywords: Credit risk, Merton Model, Distance to default.

1. Introduction

Credit risk refers to the risk that the counterparty may fail to meet its obligations in the agreed terms. Quantifiying it accurately is of immense importance to all: credit risk managers, regulators, and investors. One measure to assess the credit risk of a firm is distance to default (DtD), which is measured as the difference between the asset value of the firm and the face value of its debt, scaled by the standard deviation of firm's asset value. It captures how many standard deviations away a firm is from the default. Higher values of DtD imply lesser likelihood of the firm to default on its credit obligations. The measure is derived using the structural default model of Merton (1974) (also referred to as the *Merton Model*) which exploits the interpretation of equity as a call option on the firm's underlying assets. The model was later extended by subsequent papers (Vasicek (1984), Moody's-KMV Crosbie and Bohn (2003).

The function `dtd` from package **ifrogs** is the first implementation of the DtD routine in R.

The paper proceeds as follows: Section 2 briefly explains the Merton Model and the derivation of DtD. Section 3 discusses its implementation in R. Section 4 illustrates the function with an example. Section 5 and 6 show the accuracy and compuational efficiency of the function.

2. The Merton Model

In the Merton (1974) framework, equity is viewed as a European call option on the underlying market value of the firm's assets, with a strike price equal to the face value of its debt. The reason is that equity-holders are only the residual claimants on the firm's assets, once all obligations are met. As long as the market value of the firm's assets is greater than the face value of its debt, the payoff to the equity-holders is a positive amount. However, as soon as its assets value hits the debt value, the firm is assumed to default, and the payoff to the equity-holders is zero.

The market value of the firm's assets is the sum of the market values of its debt and equity. While the market value of the firm's equity is directly observable, the market value of the debt is not. Using a set of observable variables, the Merton Model derives the implied market value of the firm's assets and its volatility by backsolving the Black-Scholes Options pricing formula.

The model assumes fully liquid financial markets with no transactions costs and arbitrage opportunities, constant risk-free interest rate with no difference between the borrowing and lending rate and no penalty on short selling. It further assumes that the market value of the firm's underlying assets follows a Geometric Brownian Motion.

Under these assumptions, the firm's assets value follow the stochastic process:

$$dV_A = \mu V_A dt + \sigma_A V_A dz$$

where, V_A is the firm's asset value, dV_A is the change in asset value, μ and σ_A are the drift rate and volatility of firm's asset value, and dz is a standard Wiener process.

The capital structure in this framework allows for a single class of debt and equity. If F is the book value of the debt which is due at time T, then the market value of equity and the market value of assets (using the Black-Scholes formula) are related by the following expression:

$$V_E = V_A N(d_1) - e^{-rT} F N(d_2) \tag{1}$$

where, V_E is the market value of the firm's equity, V_A is the market value of assets, r is the risk free interest rate, and $N(d_1)$ and $N(d_2)$ are the standard cumulative normal of d_1 and d_2 given as:

$$d_1 = \frac{ln(\frac{V_A}{F}) + (r + \frac{\sigma_A^2}{2})T}{\sigma_A \sqrt{T}}$$

$$d_2 = d_1 - \sigma_A \sqrt{T}$$

Using Ito's lemma, the equity volatility (σ_E) and asset volatility (σ_A) are related by the expression:

$$\sigma_E = \frac{V_A}{V_E} \cdot \frac{\partial V_E}{\partial V_A} \sigma_A$$

From the Black-Scholes options pricing formula, $\frac{\partial V_E}{\partial V_A} = N(d_1)$. Thus, the above expression can be rewritten as:

$$\sigma_E = \frac{V_A}{V_E} \cdot N(d_1)\sigma_A \tag{2}$$

In practice, V_E, σ_E, F, T, r are known. To compute V_A and σ_A, one can solve the system of two non-linear equations 1 and 2 simultaneously to *minimize* the sum of the squared errors:

$$e^2 = e_1^2 + e_2^2 \tag{3}$$

where

$$e_1 = V_E - V_A N(d_1) - e^{-rT} F N(d_2)$$

and

$$e_2 = \sigma_E - \frac{V_A}{V_E} \cdot N(d_1)\sigma_A$$

Once the values of V_A and σ_A are obtained, the Distance to default (DtD) is computed as:

$$\text{DtD} = \frac{V_A - F}{V_A \cdot \sigma_A} \tag{4}$$

3. R implementation

The function `dtd` implements the above methodology. It consumes a set of four values: market value of the equity of the firm (`mcap`), its volatility (`vol`), the face value of its debt (`debt`), and the annualized interest rate (`r`). The time-to-maturity, T is assumed to be one year. The function is given as:

```
> library(ifrogs)
> str(dtd)
```

```
function (mcap, debt, vol, r)
```

Once the above set of values are plugged in, the function `dtd` minimizes Equation 3 using the 'L-BFGS-B' algorithm of the function `optim`. The initial set of values that are supplied are:

1. Assets value: Sum of market value of equity and the face value of debt, $V_A = V_E + F$
2. Assets volatility: $\sigma_A = \frac{V_E \times \sigma_A}{F}$

In addition, the lower limit for the firm's asset value and its volatility is set as the market capitalization of the firm and zero respectively. The upper limit for both the variables is set as infinity.

The function returns three set of values: DtD, estimated values of firm's assets and its volatility.

4. Example

We now illustrate the function with the help of an example. We input the `mcap` as 10000, `debt` as 5000, `vol` of equity as 0.4 and `r` as 0.10, for a hypothetical firm 'X' as follows:

```
> dtd(mcap=10000, debt=5000, vol=0.4, r=0.1)

       dtd.v      asset.v      sigma.v
2.499983e+00 1.452418e+04 2.754045e-01
```

The output `dtd.v` shows the estimated DtD value of the firm.

To compute a time-series of the DtD, we use an example dataset for a firm, Reliance Industries Ltd., listed on the National Stock Exchange, India. The dataset, `dtd_reliance.RData` has a set of three variables: E as the market value of the equity, sE as its volatility and F as the debt of the firm.

```
> data(dtd_reliance)
> head(dtd_reliance)

        date       E        sE      F
1 2011-01-04 3523435 0.2391617 311969
2 2011-01-05 3514761 0.2391564 311969
3 2011-01-06 3547818 0.2386119 311969
4 2011-01-07 3485305 0.2392654 311969
5 2011-01-10 3374518 0.2413854 311969
6 2011-01-11 3319697 0.2400812 311969
```

One can generate a time series of the firm's DtD using the function in the following manner:

```
> ans <- apply(X=dtd_reliance[ , -1], MARGIN=1,
+                         FUN=function (i) dtd(mcap=i[["E"]],
+                           vol=i[["sE"]], debt=i[["F"]], r=0.05))
> ans <- data.frame(date=dtd_reliance[ , "date"], t(ans))
> head(ans)

        date    dtd.v asset.v   sigma.v
1 2011-01-04 4.181272 3820189 0.2205835
2 2011-01-05 4.181364 3811515 0.2205364
3 2011-01-06 4.190905 3844572 0.2201940
4 2011-01-07 4.179460 3782059 0.2204917
5 2011-01-10 4.142752 3671272 0.2218739
6 2011-01-11 4.165257 3616451 0.2203809
```

```
> plot(ans[,"date"], ans[,"dtd.v"], ylab="DtD", xlab="", type="l")
```

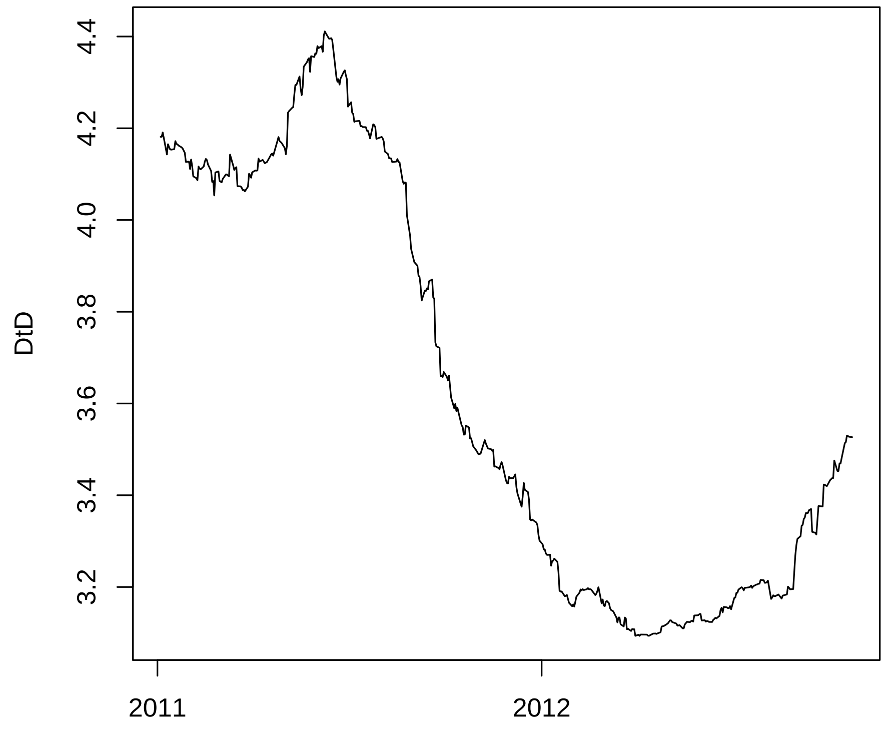

Figure 1: Reliance DtD

The values in the column `dtd.v` of `ans` show the time series of DtD. Figure 1 shows the plot of the same.

5. Benchmarking accuracy

We test the code by plugging back the values of output variables: V_A and σ_A into the Black-Scholes options pricing formula. We compare the obtained value of the option price with the market value of the equity firm (which is an input to `dtd`). In principle, the two should be the same.

As an example, we simulate 1000 values of E (market cap), sE (equity volatility) and F (debt), obtain the values of V (asset value) and sV (volatility of assets value) in the following manner:

```
> # Simulation
> compute_time <- system.time(simulate <- lapply(1:1000, function(i){
```

```
+   set.seed(i)
+   E <- rnorm(1, mean=56700, sd=200)
+   F <- rnorm(1, mean=25000, sd=400)
+   sE = rnorm(1, mean=0.25, sd=0.02)
+   ans <- dtd(mcap=E, vol=sE,
+              debt=F,  r=0.05)
+   simulate_dtd <- ans["dtd.v"]
+   simulate_v <- ans["asset.v"]
+   simulate_sv <-ans["sigma.v"]
+   return(c(E=E, sE=sE, F=F, simulate_dtd, simulate_v,
+            simulate_sv))
+ }))
> simulate_results <- do.call(rbind, simulate)
> head(simulate_results)
```

```
            E        sE        F    dtd.v  asset.v   sigma.v
[1,] 56574.71 0.2332874 25073.46 4.286558 80425.32 0.1641046
[2,] 56520.62 0.2817569 25073.94 3.549159 80371.69 0.1981428
[3,] 56507.61 0.2551758 24882.99 3.918867 80177.05 0.1798442
[4,] 56743.35 0.2678229 24783.00 3.733811 80317.67 0.1892133
[5,] 56531.83 0.2248902 25553.74 4.446615 80839.30 0.1572682
[6,] 56753.92 0.2673732 24748.01 3.740091 80294.95 0.1889842
```

We now insert the values of simulate_results$asset.V, simulate_results$sigma.v, simulate_results$F back into the Black-Scholes Options pricing formula:

```
> library(fOptions)
> Estimated.E <- GBSOption("c", S=simulate_results[,"asset.v"],
+                          X=simulate_results[,"F"],
+                          Time=1, r=0.05, b=0.05,
+                          sigma=
+                          simulate_results[,"sigma.v"])
> compare_results <- data.frame(simulate_results,
+                               Estimated.E=Estimated.E@price)
> head(compare_results)
```

```
         E        sE        F    dtd.v  asset.v   sigma.v Estimated.E
1 56574.71 0.2332874 25073.46 4.286558 80425.32 0.1641046    56574.71
2 56520.62 0.2817569 25073.94 3.549159 80371.69 0.1981428    56520.62
3 56507.61 0.2551758 24882.99 3.918867 80177.05 0.1798442    56507.61
4 56743.35 0.2678229 24783.00 3.733811 80317.67 0.1892133    56743.35
5 56531.83 0.2248902 25553.74 4.446615 80839.30 0.1572682    56531.83
6 56753.92 0.2673732 24748.01 3.740091 80294.95 0.1889842    56753.92
```

```
> all.equal(compare_results$E, compare_results$Estimated.E)
```

```
[1] TRUE
```

The result of `all.equal()` shows that the estimated values of the market cap of equity is equivalent to the actual values. This corroborates the accuracy of the code.

6. Computational efficiency

In this section, we discuss the efficiency of the code in terms of computational time taken to generate a time series of 'dtd' values. In the previous code chunk, `compute_time` records the time taken to generate 1000 values of 'dtd' using simulated data. The results are as follows:

```
> compute_time

   user  system elapsed
  2.896   0.000   2.898
```

The above computation is done on a 64bit Linux machine with R-2.15.3.

References

Crosbie PJ, Bohn JR (2003). "Modeling Default Risk." *Moody's KMV*.

Merton R (1974). "On the pricing of corporate debt: the risk structure of interest rates." *Journal of Finance*, **29**, 449–470.

Vasicek O (1984). "Credit Valuation." *KMV Corporation*.

Affiliation:

Ajay Shah
National Institute of Public Finance and Policy
Special Institutional Area
New Delhi 110067, India
Email: ajayshah@mayin.org

Manish Singh & Nidhi Aggarwal
Indira Gandhi Institute of Development Research
Goregaon East, Mumbai
400 065, India
E-mail: mks344@gmail.com, nidhi@igidr.ac.in