Fixed Income Analysis

Estimation of the Term Structure
Part II

Jesper Lund

February 17, 1998

Statistical techniques — introduction

- Data consist of N bonds, with payments b_{ij} for $i = 1, \ldots, N$ and $j = 1, \ldots, m_i$, and the respective payment dates are t_{ij}.
- Pricing equation, allowing for measurement (pricing) error ε_i

$$P_i + A_i = \sum_{j=1}^{m_i} b_{ij} \cdot d(t_{ij}) + \varepsilon_i, \qquad i = 1, 2, \ldots, N \tag{1}$$

- If $d(t)$ in (1) is parameterized using some functional form with K parameters, and $K < N$, these parameters can be estimates by non-linear regression analysis.
- Various approaches differ as to whether they parameterize $d(t)$ directly, or indirectly via spot rates $R(t)$ or forward rates $f(t)$, and which parameterization is used (often cubic spline functions).

Statistical techniques — Basic idea

- Suppose the discount factor is parameterized in terms of zero-coupon rates, that is $d(t) = \exp[-t \cdot R(t)]$, and $R(t) = R(t, \beta)$.
- There are two basic requirements for any parameterization of the yield curve function $R(t, \beta)$:
  - The function should be **sufficiently flexible**, so that (almost) any shape of the yield curve can be accommodated. Examples include monotonically increasing or decreasing, humped and inverse humped. Different values of the parameter vector β should translate into different shapes.
  - The function should be **parsimonious**, that is the number of parameters (in the vector β) should be "small". This avoids convergence problems in the estimation, and reduces the risk of overfitting "noise" in the data.
- Note that there is a (mutual) conflict between the two goals.
- Polynomials and especially (cubic) **spline functions** are often used to parameterize $R(t)$, and sometimes $d(t)$ directly.

Spline functions – 1

- A K'th order polynomial in t is defined as

$$f_K(t) = a_0 + a_1 t + a_2 t^2 + \cdots + a_{K-1} t^{K-1} + a_K t^K \tag{2}$$

- Weierstrass' theorem: by choosing a sufficiently large K, any continuous function on a closed interval — like [0, 30] — can be approximated arbitrarily well (for some constants $a_0 \dots a_K$).
- The theory is nice, but there are some practical problems:
  - A (very) high order K of the polynomial may be required in order to approximate the yield curve (function). Remember that we prefer parsimonious functions ...
  - The yield curve $R(t)$ is only observed indirectly through a limited number of bond prices. A high-order polynomial may fit these maturities quite well, but display erratic behavior between these maturities.
  - In summary: best results are obtained with a low-order polynomial on a small interval (local approximation to the function).

Spline functions – 2

- The basic idea of **spline functions** is to combine low-order polynomials (typically cubic) on different subintervals.
- The subintervals are determined by the so-called **knots** of the spline function.
- Smoothness restriction for a cubic spline: the function itself, and the first and second derivative must be continuous at the knots.
- Example: spline function on $[0, x_2]$ with two segments

$$f_1(t) = a_{10} + a_{11}t + a_{12}t^2 + a_{13}t^3, \quad t \in [0, \tau_1] \tag{3}$$
$$f_2(t) = a_{20} + a_{21}t + a_{22}t^2 + a_{23}t^3, \quad t \in [\tau_1, \tau_2] \tag{4}$$

- Restriction: $f_1(\tau_1) = f_2(\tau_1)$, $f_1'(\tau_1) = f_2'(\tau_1)$ and $f_1''(\tau_1) = f_2''(\tau_1)$.

Spline functions – 3

- The smoothness restriction reduces the number of free parameters in (3) and (4) from 8 to 5.
- In general, with s segments, there are $K = s + 3$ free parameters.
- A simple representation of the spline function on $[0, \tau_s]$ is the truncated power basis:

$$f(t) = a_1 + b_1 t + c_1 t^2 + d_1 t^3 + \sum_{i=1}^{s-1} d_{i+1}(t - \tau_i)^3 D_i, \tag{5}$$

where $D_i = 1$ if $t \geq \tau_i$ and $D_i = 0$ otherwise.

- The truncated power basis can be numerically unstable because the terms in (5) are highly correlated. Most people use **B-splines**, which represent a stable basis, but the basis functions are much more complex. See section 2.4.4 in Anderson et al. (1996).

The discount function as a cubic spline

- Introduced by McCulloch (1971, 1975) who use a stable spline representation (denoted by $\theta(t)$ here)

$$d(t) = 1 + \sum_{k=1}^{K} a_k \theta_k(t) \tag{6}$$

- If we substitute this into (1), we get

$$P_i + A_i - \sum_{j=1}^{m_i} b_{ij} = \sum_{k=1}^{K} a_k \left(\sum_{j=1}^{m_i} \theta(t_{ij}) b_{ij} \right) + \varepsilon_i \tag{7}$$

- The unknown parameters $a_1, \ldots, a_K$ can be estimates by ordinary (linear) least squares.
- The method is simple to implement (does not require numerical optimization techniques), but there are some serious limitations.

The spot curve $R(t)$ as a cubic spline – 1

- Main disadvantages of the McCulloch technique:
  - Lack of stability for spot rates $R(t)$, and especially forward rates $f(t)$, in the long end of the curve.
  - The discount function $d(t)$ is really exponential, rather than polynomial.
  - Impossible to extrapolate beyond the largest maturity of the N bonds used for estimation.
- A better approach: parameterize the spot curve $R(t)$ in terms of a cubic spline with basis $\phi_k(t)$, e.g. a B-spline basis,

$$R(t) = \sum_{k=1}^{K} a_k \phi_k(t) \tag{8}$$

- Assuming continuous compounding (convenient here), the discount function is given by $d(t) = \exp\{-t\, R(t)\}$.

The spot curve $R(t)$ as a cubic spline – 2

- In this case, we get the following regression model for bond prices:

$$P_i + A_i = \sum_{j=1}^{m_i} b_{ij} \exp \left\{ - \sum_{k=1}^{K} a_k \cdot [t_{ij} \phi_k(t_{ij})] \right\} + \varepsilon_i \qquad (9)$$

- Estimating this model requires non-linear least squares (NLS), and hence numerical optimization. With state-of-the-art computers, this is no longer a problem . . .
- If the last segment of the cubic spline is constrained to be flat, we can even use the technique for **extrapolation**, that is estimate $R(t)$ for t beyond the largest maturity of the N coupon bonds.
- Main limitation: the number of segments and position of the knots is somewhat arbitrary.
- The non-parametric **smoothing splines** avoid these problems.

Other parameterizations of $R(t)$

- Polynomial method of Chambers et al. (1984)

$$R(t) = \sum_{k=0}^{K} a_k t^k \qquad (10)$$

- This is simpler than splines, but (10) tends to lack stability for large K, and the method cannot be used for extrapolation.
- Nelson and Siegel (1987) propose the following four-parameter $(\beta_0, \beta_1, \beta_2, \tau_1)$ model for the spot curve

$$R(t) = \beta_0 + (\beta_1 + \beta_2) \left[1 - \exp\left(-\frac{t}{\tau_1} \right) \right] \frac{\tau_1}{t} - \beta_2 \exp\left(-\frac{t}{\tau_1} \right) \qquad (11)$$

- The Nelson-Siegel model is very parsimonious, but may lack the necessary flexibility in some cases.

Estimation (fitting) techniques – 1

- General setup: non-linear regression model for $P^* = P + A$,

$$\begin{aligned} P_i^* &= \sum_{j=1}^{m_i} b_{ij} \exp\left(-t \cdot R(t,\beta)\right) + \varepsilon_i \\ &\equiv Z_i(\beta) + \varepsilon_i, \quad i = 1, 2, \ldots, N \end{aligned} \tag{12}$$

- The most common estimation technique is the (weighted) non-linear least squares (NLS) method, where β is estimated by

$$\widehat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} w_i \left[P_i^* - Z_i(\beta)\right]^2 \tag{13}$$

- If $w_i = 1$ for all i, we get unweighted NLS.
- This assumes that the error term ε_i in (12) is homoskedastic (constant variance for all bonds).

Estimation (fitting) techniques – 2

- The homoskedasticity assumption is often violated by the data, and pricing errors are more dispersed for long-term bonds.
- This may be accomodated by specifications like

$$w_i = 1/T_i^{\delta} \qquad \text{or} \qquad w_i = 1/D_i^{\delta},$$

where T_i and D_i are the bond maturity and duration, and $\delta > 0$.

- Least-squares estimates can be sensitive to large residuals (outliers). A way to avoid this potential is using least absolute deviations (LAD) instead

$$\widehat{\beta} = \arg\min_{\beta} \sum_{i=1}^{N} \left|P_i^* - Z_i(\beta)\right| \tag{14}$$

- LAD estimates are more robust, but much harder to compute than NLS.