

收益率曲线的构建

郭宜昕

摘要. 本文简要回顾了一些常见的收益率曲线构建方法，并且进一步讨论了 Nelson-Siegel 模型在收益率曲线估计上的可能应用。

1. 构建收益率曲线的理论回顾

1.1. 对市面上常见构建收益率曲线方法的回顾.

市面上构建收益率曲线的方法常见以下三种：

1. 以 YY 评级为例，基于最小二乘法使用全量数据构建收益率曲线。
2. 以中债、中证收益率曲线为例，基于选择特定剩余期限的关键点，针对以上关键点附近数据进行插值构建收益率曲线。
3. 基于 Nelson-Siegel 模型，

	中债估值	中证估值	YY 估值
样本选择	债券评级、债券性质、债券条款、债券流动性等因素，各家机构对于样本券的选取规则差异较大，YY 估值按照 YY 评级选取资质稳定、成交活跃的债券作为样本。		
收益率曲线算法	三次 Hermite 插值法	贝叶斯平滑样条模型	最小二乘法曲线拟合
曲线及隐含评级的调整	样本券选择不同，成交数据的清洗和采用规则不同，影响曲线形状和主体资质的程度不同		
市场成交数据的选取	银行间双边报价、结算数据、柜台市场双边报价、交易所债券交易数据以及市场成员的收益率估值数据	银行间、交易所、经纪商报价、成交行情、货币市场数据、债券股票一、二级市场数据及市场成员估值	银行间、交易所、中介成交、中介报价数据和债券的一级发行数据

图 1. 目前市面上的收益率曲线构建方法

1.2. 利率期限结构.

利率期限结构指某个时间点上，不同期限的即期利率（或到期收益率）和到期期限之间的关系。具体而言，我们可以将其分为几种常见的利率期限结构形态：

一般来说，收益率曲线有以下三个特征：

- 不同期限的债券收益率有同向运动的趋势；
- 收益率曲线通常倾向于向上倾斜；
- 短期债券收益率的波动通常要比长期债券收益率的波动大；

MR (2020) 主题分类. 05A19 .

关键词. 收益率曲线，插值多项式，Nelson-Siegel 模型.

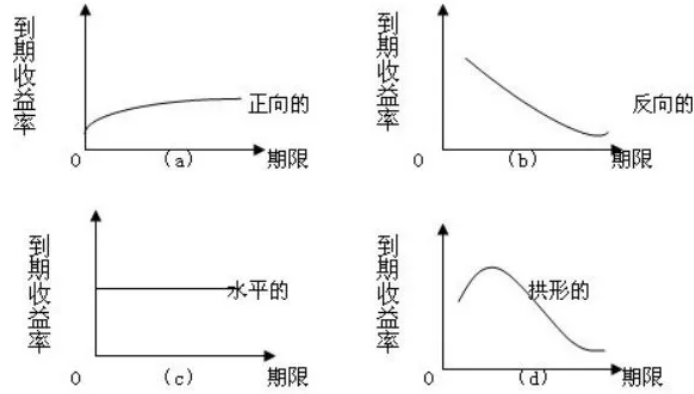


图 2. 常见的利率期限结构

1.3. Nelson-Siegel 模型.

Nelson-Siegel 发现利用 laguerre 函数族 $(1, e^{-x}, xe^{-x}, x^2e^{-x}, \dots)$ 的线性组合可以较全面的描述远期利率的几种利率期限结构, 并且其函数图像符合收益率曲线的具体特征。进一步地, 根据远期利率 $f(t)$ 与即期利率 $s(t)$ 的关系:

$$s(t) = \frac{1}{t} \int_0^t f(s) ds$$

我们就得到了通过不同形态期限结构加总得到的即期利率 $s(t)$ 。

一般地, 我们取

$$f_t(\tau) = \beta_{1t} + \beta_{2t}e^{-\tau/\lambda_t} + \beta_{3t}\lambda_t e^{-\tau/\lambda_t}$$

$$y_t(\tau) = \beta_{1t} + \beta_{2t} \left(\frac{1 - e^{-\tau/\lambda_t}}{\tau/\lambda_t} \right) + \beta_{3t} \left(\frac{1 - e^{-\tau/\lambda_t}}{\tau/\lambda_t} - e^{-\tau/\lambda_t} \right)$$

分别作为远期与即期利率的表达式。在即期利率中, 三个函数分别对应三个形态的期限结构:

- $h(t) = 1$: 对应水平的期限结构, 即水平因子。
- $h(t) = \left(\frac{1 - e^{-\tau/\lambda_t}}{\tau/\lambda_t} \right)$: 对应单调倾斜的期限结构, 即斜率因子。
- $h(t) = \left(\frac{1 - e^{-\tau/\lambda_t}}{\tau/\lambda_t} - e^{-\tau/\lambda_t} \right)$: 对应中部隆起的期限结构, 即曲率因子。

当 τ 趋近于 ∞ 时, $\lim_{\tau \rightarrow \infty} y_t(\tau) = \beta_{1t}$ 。因此 β_{1t} 代表了利率的长期水平, β_1 的变动会使整条曲线的高度上下变动, 因而被称为水平因子。当 τ 趋近于 0 时, $\lim_{\tau \rightarrow 0} y_t(\tau) = \beta_1 + \beta_2 \implies \lim_{\tau \rightarrow \infty} y_t(\tau) - \lim_{\tau \rightarrow 0} y_t(\tau) = -\beta_2$ 。 β_2 被称为斜率参数, 主要对短期利率产生影响。

远期函数的曲率因子在 $\tau = \lambda_t$ 时最大, 故我们认为 λ_t 控制曲率参数的指数增长率。 λ 越小, 指数项的衰减越慢, 对于长期限的数据点拟合越好; λ_t 越大, 指数项的衰减越快, 对于短期限的数据点拟合越好。

在实际进行数值分析时, 我们注意到原始的 Nelson-Siegel 模型在数值上存在依赖初值的选取, 参数不稳定等诸多问题。故在 2007 年, Diebold 和 Li 针对国债收益率曲线的计算提出了一种针对 Nelson-Siegel 模型的改善方法, 即 Diebold-Li 模型。该模型不再将所有的参数交由数值方法确定, 而是事先给定有关期限结构的参数 λ_t 。在相关论文里, λ_t 的确定方法为: 给定剩余期限的均值 T , 通过最大化即期函数曲率因子在 T 点的值实现, 我们在实际操作中也参考这一思路。

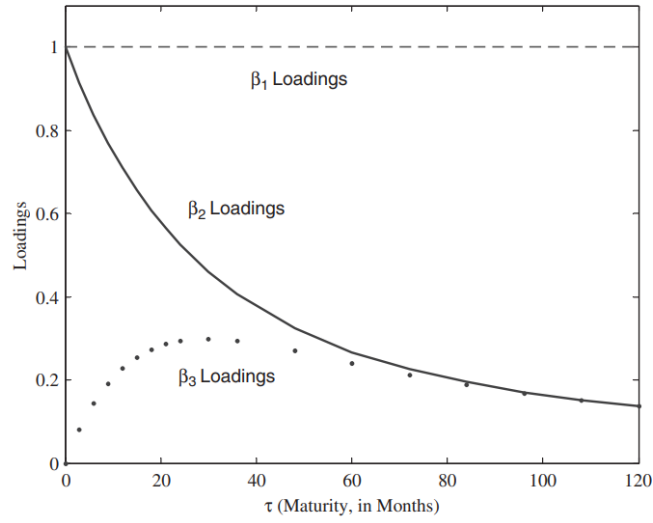


图 3. 三个因子的可视化描述

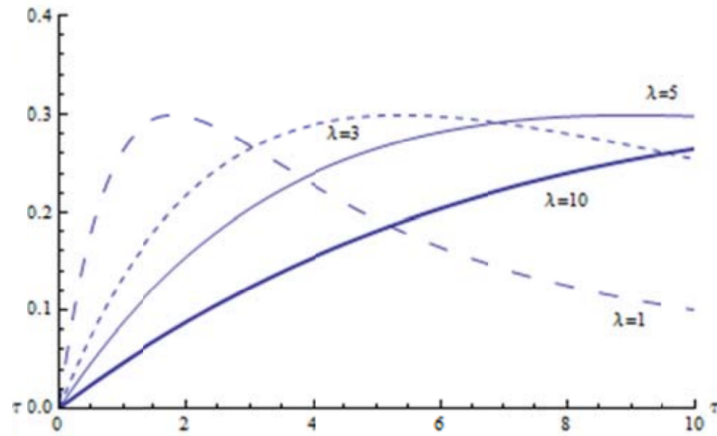


图 4. 在即期函数中曲率因子随 λ_t 的变化情况

2. 对基于 NELSON-SIEGEL 方式构建收益率曲线的展望

由于我们研究的是不同评级信用债的利率期限结构问题，故我们需要对于模型参数在信用角度体现的意义进行解释说明。

2.1. 参数 β_1 .

参数 β_1 衡量了远端的收益率水平，因此，对应于不同评级的收益率曲线，我们可以给出每个评级所对应的远端收益率情况，作为每个评级风险情况的参考。在理想情况下，风险越高的债券远端收益率水平相应的也越高，基于 DM 对城投主体的评分情况，这一现象得到了较好的支持。

2.2. 参数 β_2 .

对于资质较好主体发布的低风险债券，投资者不担心债券的兑付问题而主要关注自己面临的利率风险与再投资风险，故在持有更长期限的债券时要求更高的风险溢价水平，因此利率期限结构呈现出一般的期限越长收益率越高的特点。为了保证这一特点，我们人为的对参数 β_2 施加了限制。

对于资质较差的主体，能否成功发行并出售债券可以用信号博弈模型进行简单描述。此时投资者无法直接观察到主体的风险水平，但主体通过发行长期债券可以向投资者传递信号。而投资者在观察到公司发行长期债券的信号后，可能会解读为公司风险水平已经降低，从而更愿意购买这些债券。这将提高长期债券的价格，从而降低长期债券的收益率。这就导致了长期的收益率上行不够明显，在具体数值上反应为斜率因子的系数 β_2 相较于评级更好的债券绝对值更小。

3. 参考文献

- [1] Filipović, D. (1999), A Note on the Nelson–Siegel Family. *Mathematical Finance*, 9: 349–359. <https://doi.org/10.1111/1467-9965.00073>
- [2] Francis X. Diebold, Canlin Li, Vivian Z. Yue, Global yield curve dynamics and interactions: A dynamic Nelson–Siegel approach, *Journal of Econometrics*, Volume 146, Issue 2, 2008, Pages 351–363, ISSN 0304-4076, <https://doi.org/10.1016/j.jeconom.2008.08.017>.
- [3] De Pooter M (2007). "Examining the Nelson-Siegel Class of Term Structure Models: In-Sample Fit versus Out-of-Sample Forecasting Performance." SSRN eLibrary. <http://ssrn.com/paper=992748..>
- [4] Siegel, A.F., Nelson, C.R., 1988. Long-term behavior of yield curves. *Journal of Financial and Quantitative Analysis* 23, 105–110

4. 所用程序代码

```

1 # Define the Nelson-Siegel model with lambda as a parameter
2 def diebold_li_fixed_lambda(t, beta0, beta1, beta2, lambda_):
3     return (beta0
4             + beta1 * (1 - np.exp(-t / lambda_)) / (t / lambda_)
5             + beta2 * ((1 - np.exp(-t / lambda_)) / (t / lambda_) - np.exp(-t / lambda_)))
6
7 # Function to optimize
8 def objective_function(lambda_t, tau):
9     return -((1 - np.exp(-tau / lambda_t)) / (tau / lambda_t) - np.exp(-tau / lambda_t))
10
11
12 # Calculate average remaining tenor for each rating
13 average_remaining_tenor = filtered_data.groupby('rating')['remaining_tenor'].mean()
14 # Initialize a dictionary to store the optimal lambda for each rating
15 optimal_lambda_dict = {}
16
17 # Find the optimal lambda for each rating
18 for rating, tau in average_remaining_tenor.items():
19     res = minimize(objective_function, 1.0, args=(tau), method='Nelder-Mead')
20     optimal_lambda_dict[rating] = res.x[0]

```

```

21
22
23 # Fit the Diebold-Li model and plot the fitted curve for each rating
24 for rating in ratings:
25     # Filter the data for the current rating
26     rating_data = filtered_data[filtered_data['rating'] == rating]
27
28     # Filter out outliers
29     rating_data = rating_data[rating_data['inlier'] == True]
30
31     # Get the remaining tenor and yield data
32     x = rating_data['remaining_tenor'].values
33     y = rating_data['yield'].values
34
35     # Estimate the parameters with the optimal lambda
36     initial_guess = [np.mean(y), np.std(y), -0.01]
37     bounds = ([-np.inf, -np.inf, -np.inf], [np.inf, np.inf, 0])
38     try:
39         params, _ = curve_fit(lambda t, beta0, beta1, beta2: diebold_li_fixed_lambda(t, beta0,
40         beta1, beta2, optimal_lambda_dict[rating]), x, y, p0=initial_guess, bounds=bounds)
41     except RuntimeError:
42         params = None
43
44     # Store the optimal parameters for the current rating
45     optimal_params_dict_fixed_lambda[rating] = params
46
47     if params is not None:
48         fitted_y = diebold_li_fixed_lambda(x, *params, optimal_lambda_dict[rating])
49         r2 = r2_score(y, fitted_y)
50         r2_scores[rating] = r2
51
52     # Plot the fitted curve
53     if params is not None:
54         t = np.linspace(min(x), max(x), 500)
55         ax.plot(t, diebold_li_fixed_lambda(t, *params, optimal_lambda_dict[rating]), label=f'Rating
56         {rating}')

```

LISTING 1. Nelson-Siegel 模型