Chapter 1

Python 金融分析概述

1.1 对时间序列数据进行金融分析

1.1.1 下载多个时间序列数据

默认情况下，`quandl.get()` 返回的是每日价格，我们还可以指定数据集下载其他类型的数据，比如，示例中的 `collapse='monthly'` 代码表明下载的是每月价格。

1.1.2 显示相关矩阵

1.1.3 指数移动平均

SMA 和 EMA 的图模式大致相同，但由于 EMA 对近期数据的权重高于较早数据的权重，因此它对价格变化的反应会比 SMA 更敏感。

Chapter 2

金融中的线性问题

2.1 资本资产定价模型与证券市场线

2.2 LU 分解

LU 分解，也称为上下因子分解（lower upper factorization），是一种线性方程组的求解方法。LU 分解将矩阵 A 分解为两个矩阵的乘积：一个下三角矩阵 L 和一个上三角矩阵 U。分解过程如下所示：

$$A = LU$$

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} = \begin{bmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{22} & 0 \\ l_{31} & l_{32} & l_{33} \end{bmatrix} \times \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix} \tag{2.1}$$

矩阵 A 中，$a = l_{11}u_{11}$，$b = l_{11}u_{12}$，以此类推。下三角矩阵对角线右上方系数全部为零，相反即为上三角矩阵。LU分解可应用于任意方阵。

2.3 Cholesky 分解

Cholesky 分解是利用对称矩阵性质求解线性方程组的方法。与 LU 分解相比，它可以显著提高计算速度并降低对内存的要求。但使用 Cholesky 分解时需要矩阵为埃尔米特矩阵（实值对称矩阵）且正定，即Cholesky分解将矩阵分解为$A = LLT$，其中 L 是对角线为正实数的下三角矩阵，L^T 为 L 的共轭转置矩阵。

2.4 QR 分解

QR 分解，又称为 QR 因式分解，和 LU 分解一样是利用矩阵求解线性方程组的方法。QR 分解用于处理 $Ax = B$ 形式的方程，且矩阵 $A = QR$，Q 为正交矩阵，R 为上三角矩阵。QR 算法是线性最小二乘问题的常见解法。

一个正交矩阵具有下列特征：

- 它是一个方阵。
- 正交矩阵乘以其转置矩阵等于单位矩阵： $QQ^T = Q^TQ = 1$
- 正交矩阵的逆矩阵等于其转置矩阵： $Q^T = Q^{-1}$

单位矩阵是一个方阵，其主对角线上元素均为 1，其余全为 0。现在将 $Ax = B$ 的问题转化成：

$$\begin{aligned} QRx &= B \\ Rx &= Q^{-1}B \\ Rx &= Q^TB \end{aligned} \tag{2.2}$$

2.5 使用其他矩阵代数方法求解

某些情况下，我们求的解不收敛，可以使用迭代法解决此类问题，如 Jacobi 迭代、Gauss-Seidel 迭代和 SOR 迭代法。

2.5.1 Jacobi 迭代法

Jacobi 迭代法通过对矩阵的对角元迭代求解线性方程组，计算结果收敛时终止迭代。方程 $Ax = B$ 中，矩阵 $A = D + R$，矩阵 D 为对角矩阵。

通过迭代得出答案：

$$\begin{aligned} Ax &= B \\ (D+R)x &= B \\ Dx &= B–Rx \\ x_{n+1} &= D^{-1}(B–Rx_n) \end{aligned} \tag{2.3}$$

与 Gauss-Seidel 迭代法不同，Jacobi 方法欲求 x_{n+1} 必须先求出 x_n，这将占用两倍的内存。然而，矩阵中每个元素的计算都以并行方式完成会显著提高运算速度。如果矩阵 A 是一个不可约严格对角占优（strictly irreducibly diagonally dominant）矩阵，通过 Jacobi 迭代法得到的解一定是收敛的。不可约严格对角占优矩阵是每个对角元的绝对值都大于所在行非对角元绝对值之和的矩阵。

2.5.2 Gauss-Seidel 迭代法

Gauss-Seidel 迭代法与 Jacobi 迭代法很相似。方程 $Ax = B$ 中，矩阵 $A = L + U$，矩阵 L 为下三角矩阵，矩阵 U 为上三角矩阵。迭代得：

$$\begin{aligned} Ax &= B \\ (L+U)x &= B \\ Lx &= B–Ux \\ x_{n+1} &= L^{-1}(B–Ux_n) \end{aligned} \tag{2.4}$$

利用下三角矩阵 L 计算 x_{n+1}，不必先求出 x_n，这相比 Jacobi 迭代法将节省一半的存储空间。

利用 Gauss-Seidel 迭代法求解的收敛速度很大程度取决于矩阵性质，需要严格对角占优或正定矩阵。即使这些条件没有满足，Gauss-Seidel 迭代的结果仍可能收敛。

Chapter 3

金融中的非线性问题

金融行业从业者使用非线性模型预测波动性、衍生价格和计算风险价值（Value at Risk，VAR）。与线性模型求解不同，非线性模型不一定推断出全局最优解。数值求根方法通常收敛到最近的局部最优解，即方程只有一个根。

3.1 非线性建模

由于非线性研究太过宽泛和深入，本节将简要介绍实际中常用的一些非线性模型：隐含波动率模型、马尔可夫转换模型（Markov switching model）、门限模型（threshold model）和平滑转换模型（smooth transition model）。

3.1.1 隐含波动率模型

最常见的期权定价模型应该是布莱克–斯克尔斯–默顿期权定价模型（Black-Scholes-Merton model），简称为布莱克–斯克尔斯模型。布莱克–斯克尔斯模型假设证券收益服从正态分布，或证券价格服从对数正态分布，从而计算期权平价。

该模型采用以下假设变量：行权价格（K）、到期日（T）、无风险利率（r）、潜在收益波动率（σ）、标的资产当前价格（S）、收益（q）。看涨期权的数学公式 $C(S,t)$ 表示为：

$$C(S,t) = Se^{qT}N(d_1) – Ke^{-rT}N(d_2) \tag{3.1}$$

上式中：

$$d_1 = \frac{\ln(S/K) + (r - q + \sigma^2)T}{\sigma\sqrt{T}}$$

由于市场调节作用，期权价格可能与布莱克–斯克尔斯模型计算结果有偏差。特别地，实际波动率（即通过历史市场价格得到的标的收益波动性）可能与该模型采用的波动率 σ 不一致。随着波动率在证券定价中的重要性日渐提高，诸多波动率模型被提出，隐含波动率模型即其中之一。

3.1.2 马尔可夫机制转换模型

马尔可夫机制转换模型（Markov regime-switching model）又称为马尔可夫转换模型，用于金融时间序列的非线性模型构造，可在不同机制状态下描述时间序列。

3.1.3 门限自回归模型

与马尔可夫机制转换模型十分相似的门限自回归模型（Threshold Autoregressive Model，TAR）是用于解释非线性时间序列问题最常见的自回归模型。使用回归方法，简单的 AR 模型可以说是解释非线性行为的最流行模型。该门限模型的机制由时间序列过去的 d 值确定，与阈值 c 有关。

下面是自激励门限自回归（SelfExciting TAR，SETAR）模型的示例。自激励门限自回归模型可根据以往时间序列取值在不同机制间转换。

$$y_t = \begin{cases} a_1 + b_1 y_{t-d} + \varepsilon_t, & y_{t-d} \le c \\ a_2 + b_1 y_{t-d} + \varepsilon_t, & y_{t-d} \ge c \end{cases} \tag{3.2}$$

3.1.4 平滑转换模型

机制状态的快速转换在现实世界中不可能发生，因此我们要引入一个平滑连续函数。通过逻辑函数 $G(y_{t-1};\gamma,c)$，自激励门限自回归模型就变为逻辑平滑转换门限自回归（Logistic Smooth Transition Threshold Autoregressive，LSTAR）模型：

$$G(y_{t-1};\gamma,c) = \frac{1}{1+\exp\{-\gamma(y_{t-d}-c)\}} \tag{3.3}$$

自激励门限自回归模型变为逻辑平滑转换门限自回归，后者可表示为：

$$y_t = (a_1 + b_1 y_{t-d})(1\text{–}G(y_{t-1};\gamma,c)) + (a_2 + b_2 y_{t-d})G(y_{t-1};\gamma,c) + \varepsilon_t \tag{3.4}$$

上式中，参数 γ 控制机制状态的转变。γ 越大，转换越快，且 y_{t-d}越靠近阈值 c。$\gamma = 0$ 时逻辑平滑转换门限自回归模型相当于一个简单的单机制自回归模型。

3.2 非线性模型求根算法

3.2.1 增量法

增量法（incremental search）是求解非线性函数的简略方法。对于任意起点 a，可以针对每个增量 dx 得到$f(a)$ 的值。假定对于增量 dx，$f(a+dx)$、$f(a+2dx)$、$f(a+3dx)\ldots$ 的符号相同。函数值符号改变时得到解，迭代结束；如果超过边界点 b 没有得到合适解，迭代也结束。

3.2.2 二分法

二分法最大的优势为在给定的迭代次数和允许误差内，一定可以收敛得到根的近似解。某些连续函数求导尤为复杂，而二分法不要求对未知函数求导，处理非平滑函数时非常有效。

与其他求根方法相比，二分法主要的缺陷为迭代时间更长。由于二分法的求根范围在a与b之间，因此需要准确估测根的位置，否则可能会得到错误的解甚至无解。采用更大范围的区间，所需迭代时间更长。

二分法可以稳定收敛且初始无须估计近似根，通常将其结合其他方法使用，例如牛顿迭代法，更快速地获得精确结果。

3.2.3 牛顿法

3.2.4 割线法

3.3 利用 SciPy 求根

3.3.1 通用非线性求解器

scipy.optimize 模块还包含多维通用求解器，其中 root 和 fsolve 函数具有如下特征：

1. `root(fun, x0[, args, method, jac, tol, ...])`：求一个向量函数的根。
2. `fsolve(func, x0[, args, fprime, ...])`：求一个函数的根。

返回的输出作为字典对象。

Chapter 4

4.1

4.1.1 Cox-Ross-Rubinstein 模型

Cox-Ross-Rubinstein（CRR）模型提出，短期风险中性的环境下，二叉树模型与标的股票的均值和方差相匹配。标的股票波动率或股票收益的标准差的表达式如下：

$$\begin{aligned} u &= \exp(\sigma\sqrt{\Delta t}) \\ d &= \frac{1}{u} = \exp(-\sigma\sqrt{\Delta t}) \end{aligned} \tag{4.1}$$

4.1.2 Leisen-Reimer 模型

4.2 希腊值