

数学实验 Project 2

2021 年 12 月 26 日

1 问题 1

1.1 第一问

首先写一个生成高斯勒让德求积的系数和根的函数，即 Gauss.m, 通过输入整数 N 生成对应的系数和根。再生成 P_i , 最后求出所有的 P 的和。整个代码在 exp1_1.m 中。

通过计算得到对于所有的 n, $\sum_{i=1}^{n+1} P_i = 1$. 下面给出证明:

$$\sum_{i=1}^{n+1} P_i = \sum_{i=1}^{n+1} \frac{W_i \sum_{j=0}^n \phi_j^2(x_i)}{n+1} = \frac{\sum_{i=1}^{n+1} W_i \sum_{j=0}^n \phi_j^2(x_i)}{n+1} \quad (1)$$

把分母单独提出来, 即证分母等于 $n+1$ 即可。交换两求和符号顺序得

$$\sum_{i=1}^{n+1} W_i \sum_{j=0}^n \phi_j^2(x_i) = \sum_{j=0}^n \sum_{i=1}^{n+1} W_i \phi_j^2(x_i) \quad (2)$$

注意到里面的求和即是对每个 ϕ_j 作 $[0,1]$ 上的高斯求积, 即值为 1. 而外面的求和做了 $n+1$ 次, 因此分母的和为 $n+1$. 所以最终求得 $\sum_{i=1}^{n+1} P_i = 1$. \square

1.2 第二问

观察需要优化的式子即可知道这是一个二次型最优化, 而二次型优化的标准式为

$$\min_x \frac{1}{2} x^T H x + f^T x \quad (3)$$

则我们只需要算出二次型优化中的 \mathbf{H}, \mathbf{f} 即可.

$$\sum_{i=1}^{n+1} \left| \sum_{j=0}^n c_j \phi_j(x_i) - f(x_i) \right|^2 = |\Phi \hat{\mathbf{c}} - \mathbf{F}|^2 \quad (4)$$

这里 Φ 的每个元素为 $\Phi_{ij} = \phi_{j-1}(x_i)$, 且 Φ 为 $(n+1) \times (n+1)$ 的矩阵。 $\hat{\mathbf{c}}, \mathbf{F}$ 分别代表系数与每个 x_i 对应的 f 的值。

把上面的式子变为二次型标准型:

$$|\Phi\hat{\mathbf{c}} - \mathbf{F}|^2 = (\Phi\hat{\mathbf{c}} - \mathbf{F})'(\Phi\hat{\mathbf{c}} - \mathbf{F}) = (\hat{\mathbf{c}}'\Phi' - \mathbf{F}')(\Phi\hat{\mathbf{c}} - \mathbf{F}) = \hat{\mathbf{c}}'\Phi'\Phi\hat{\mathbf{c}} - 2\mathbf{F}'\Phi\hat{\mathbf{c}} + \mathbf{F}'\mathbf{F} \quad (5)$$

则令 $\mathbf{H} = \Phi'\Phi$, $\mathbf{f} = -\Phi'\mathbf{F}$. 在代码中用循环生成 Φ , 把它存在 legendre_matrix 中。之后相继生成 \mathbf{H} , \mathbf{f} , 并存在 \mathbf{H} 和 $\mathbf{f_value}$ 中。用 'quadprog' 函数对其进行优化, 即可得到 $\hat{\mathbf{c}}$ 。

将 c_k 与 $\sum_{i=1}^{n+1} f(x_i) w_i \phi_k(x_i)$ 进行比较, 发现

$$c_k = \sum_{i=1}^{n+1} f(x_i) w_i \phi_k(x_i), \quad k = 0, 1, \dots, n \quad (6)$$

相关程序在 exp1_23.m 中。下面给出数学证明。

首先注意到 \mathbf{H} 必定为正定矩阵, 因为 $\Phi'\Phi$ 的所有特征值必定都大于 0. 则该二次型的最优解为 $\hat{\mathbf{c}} = \mathbf{H}^{-1}(-\mathbf{F})$, 即 $\hat{\mathbf{c}} = \Phi'\Phi^{-1}\Phi'\mathbf{F}$. 接下来只需证明 $\Phi'\Phi^{-1}\Phi'\mathbf{F} = \Phi'\text{diag}(\mathbf{W})\mathbf{F}$ 即可。其中 $\text{diag}(\mathbf{W})$ 指的是对角线全为 \mathbf{W} 的元素的 $(n+1) \times (n+1)$ 的矩阵。

可以验证 $\Phi'\text{diag}(\mathbf{W})\mathbf{F}$ 的第 k 个元素为 $\sum_{i=1}^{n+1} f(x_i) w_i \phi_k(x_i)$ 。

下面接着证明 $\Phi'\Phi^{-1}\Phi'\mathbf{F} = \Phi'\text{diag}(\mathbf{W})\mathbf{F}$ 。

$$\begin{aligned} \Phi'\Phi^{-1}\Phi'\mathbf{F} &= \Phi'\text{diag}(\mathbf{W})\mathbf{F} \\ \Phi'\mathbf{F} &= \Phi'\Phi\Phi'\text{diag}(\mathbf{W})\mathbf{F} \\ \mathbf{F} &= \Phi\Phi'\text{diag}(\mathbf{W})\mathbf{F} \\ \mathbf{I} &= \Phi\Phi'\text{diag}(\mathbf{W}) \end{aligned} \quad (7)$$

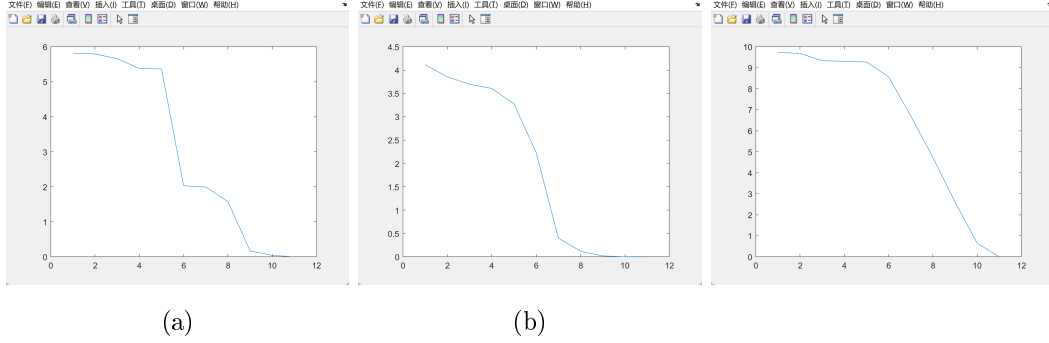
第三个等式能消去 Φ' 的原因是因为可逆, 第四个等式能消去 \mathbf{F} 的原因是 $\Phi, \text{diag}(\mathbf{W})$ 都与 \mathbf{F} 无关且 \mathbf{F} 可以几乎取任意的值 (由函数 $f(x)$ 而定)。

最后 $\Phi\Phi'\text{diag}(\mathbf{W})$ 即是 $\int_{-1}^1 \phi_i(x)\phi_j(x)dx$. 则对角元上 $i = j$, 为 1, 非对角元上 $i \neq j$, 为 0. 因此, 最后等于单位矩阵, 因此, 上式的第四个等式成立, 从而第一个式子成立, 即猜想正确。 \square

1.3 第三问

紧接着上述的程序 (即相关代码在 exp1_23.m 中)。 Φ_i 即是 Φ 的第 i 行的数据, 即 legendre_matrix 的第 i 行。之后开始每次迭代, 把每次计算出的误差存入变量 \mathbf{e} 中。这

里并没有计入第一次随机生成 \mathbf{c} 时的误差, 因此 \mathbf{e} 为 $(n+1) \times 1$ 的向量。下面分别是 $f(x) = x$, $f(x) = \sin(x)$, $f(x) = e^x$ 的误差图像。(此时 n 都取的是 10)



可以看出误差都呈下降趋势, 且当 $i = n+1$ 时, 误差非常小, 达到了 10^{-20} 的数量级或者更小。因此, 猜想是当 $i = n+1$ 时, $\mathbf{C}^{(i)}$ 几乎达到真实值 $\hat{\mathbf{c}}$ 。

下面给出证明:

注意到递推式与牛顿法的形式很相近, 因此考虑用牛顿法证明该数列收敛且收敛到 $\hat{\mathbf{c}}$ 。牛顿法的原始形式为

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \quad (8)$$

令 $g(\hat{\mathbf{c}}) = (\mathbf{c} \cdot \Phi_i - f(x_i))\Phi_i$, 则 $\frac{\partial g}{\partial \hat{\mathbf{c}}} = \Phi_i \cdot \Phi_i = \|\Phi_i\|^2$, 正好等于分母。

因此该递推式收敛的条件为分子有零点, 即

$$g(\hat{\mathbf{c}}) = (\mathbf{c} \cdot \Phi_i - f(x_i))\Phi_i \quad (9)$$

有零点。

注意到

$$\hat{\mathbf{c}} = \Phi' \Phi^{-1} \Phi \mathbf{F}'$$

两边同时左乘 Φ 得

$$\Phi \hat{\mathbf{c}} = \Phi \Phi' \Phi^{-1} \Phi \mathbf{F}'$$

因为 Φ 是可逆的, 因此括号中的项可以提出, 得

$$\Phi \hat{\mathbf{c}} = \Phi \Phi^{-1} \Phi'^{-1} \Phi' \mathbf{F} \Phi \hat{\mathbf{c}} = \mathbf{F}$$

所以 \mathbf{F} 的第 i 个元素为 Φ 的第 i 行乘以 $\hat{\mathbf{c}}$, 即 $\Phi_i \cdot \hat{\mathbf{c}}$ 。

所以(9)式有解, 也就是说 $\mathbf{c}^{(i)}$ 收敛于 $\hat{\mathbf{c}}$ 。

2 问题 2

因为每一年可用的投资数都可以根据前一年的本利来投资，因此，可以列出每一年的可用投资数，实际投资数，以及剩余资本数。这三个量满足 可用投资数 - 实际投资数 = 剩余资本，且第一年的可用投资数为 $100w$ 。

我们需要优化的目标函数就是第五年年末的本利，优化的参数则是每个项目每年的实际投资数。因为共计投资 11 次，因此是一个 11×1 的向量。其中 $x_1 \sim x_4$ 为项目一在 1 到 4 年的投资数。 x_5, x_6 分别代表项目 2 和项目 3 在第 3 年和第 2 年的投资数。 $x_7 \sim x_{11}$ 代表项目 5 在第 1 到 5 年的投资数。需要的约束条件有下界，即全部都大于等于 0。约束条件中的上界除了 x_5, x_6 设为 40 和 30，其余设为 100 即可。最后在每次投资时不得超出可用的投资数，即每次的剩余资本都要大于等于 0。这个可以另写一个函数用作非线性约束。

优化的目标函数在 exp2func1.m 中，非线性优化约束在 exp2func2.m 中。主程序在 exp2.m 中。

以下是优化的结果

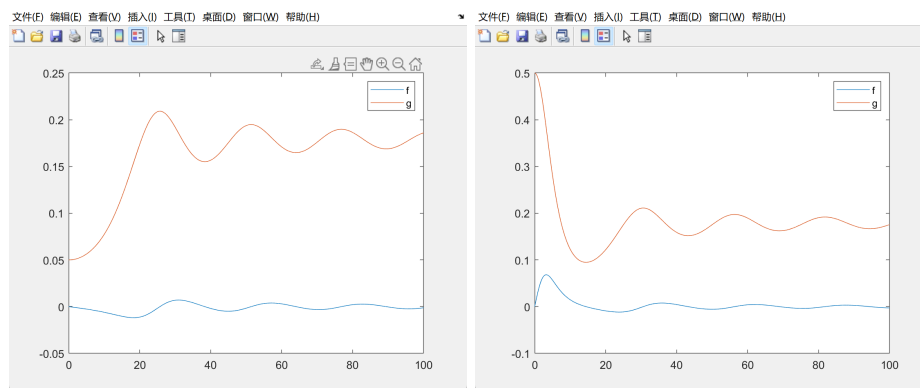
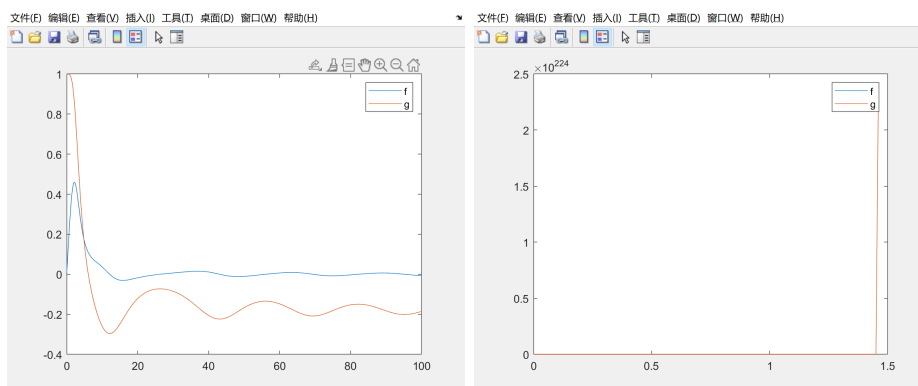
```
55.8085
16.8429
16.0904
27.9442
40.0000
30.0000
44.1915
0.0000
8.0895
0.0000
18.5039

-143.7500
```

因此第五年年末能收获的最大本利为 $143.75w$ 。

3 问题 3

依照题目所给的迭代式计算，相关代码在 `exp3.m` 中，以下分别是 g_0 取 0.05, 0.5, 1, 1.5 的时候的图像， r 取值在 0 到 100 之间的图像。

(a) $g_0 = 0.05$ (b) $g_0 = 0.5$ (c) $g_0 = 1$ (d) $g_0 = -0.5$

应用这个方法的有点有如下几点：

- 这种方法可以避免初始值奇异的情况，注意到在原表达式中存在 f/r 的项，则在做第一次迭代时会出现正无穷，而用 $r_{n+\frac{1}{2}}$ 则可以避免这种情况。
- 注意到 $\frac{1}{r^2} \frac{d(r^2 f)}{dr} = \frac{df}{dr} + \frac{2f}{r}$ ，这种方法可以使得计算 f 时具有更高的精度。