手写VIO第二章作业讲解

主讲人 啦啦啦

题目

深蓝学院 shenlanxueyuan.com

作业

1 样例代码给出了使用 LM 算法来估计曲线 $y = \exp(ax^2 + bx + c)$ 参数 a, b, c 的完整过程。

1. 请绘制样例代码中 LM 阻尼因子 μ 随着迭代变化的曲线图
2. 将曲线函数改成 $y = ax^2 + bx + c$，请修改样例代码中残差计算，雅克比计算等函数，完成曲线参数估计。
3. 实现其他更优秀的阻尼因子策略，并给出实验对比（选做，评优秀），策略可参考论文[a] 4.1.1 节。

2 公式推导，根据课程知识，完成 $\mathbf{F}, \mathbf{G}$ 中如下两项的推导过程：

$$\mathbf{f}_{15} = \frac{\partial \boldsymbol{\alpha}_{b_i b_{k+1}}}{\partial \delta \mathbf{b}_k^g} = -\frac{1}{4}(\mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)]_\times \delta t^2)(-\delta t)$$

$$\mathbf{g}_{12} = \frac{\partial \boldsymbol{\alpha}_{b_i b_{k+1}}}{\partial \mathbf{n}_k^g} = -\frac{1}{4}(\mathbf{R}_{b_i b_{k+1}}[(\mathbf{a}^{b_{k+1}} - \mathbf{b}_k^a)]_\times \delta t^2)(\frac{1}{2}\delta t)$$

3 证明式(9)。

[a] Henri Gavin. "The Levenberg-Marquardt method for nonlinear least squares curve-fitting problems". In: *Department of Civil and Environmental Engineering, Duke University* (2011), pp. 1–15.

纲要

- 第一部分：概述
- 第二部分：方法

  ****

  ****
- 第三部分：问题与挑战

作业情况

- T1.1： 绘制阻尼因子曲线。 涉及数据保存到文件，以及对文件内数据进行读取和操作。
- T1.2：曲线函数改变。考察对残差和雅克比。
- T1.3：实现其他优秀更新策略。作业中相对较难的，考察对论文与代码结合的程度。1.阅读并总结论文对应步骤 2.在代码中复现
- T2：考察求导内容　　T3：证明公式

纲要

- 第一部分：概述
- **第二部分：方法**
  - ********************
  - ********************
- 第三部分：问题与挑战

第一题

- T1.1：找到problem.cpp中的Problem::Solve()函数， 相应代码即在该函数内。可以输出两种Lambda（也即μ）

  - ```
    1  while (!stop && (iter < iterations))
    ```

  一种是在外层while中尝试的 Lambda。（所有的Lambda）

  - ```
    1  while (!oneStepSuccess)
    ```

  另一种是在 `IsGoodStepInLM()` == true中更新的Lambda。（使得误差下降的）

第一题

```
std::cout << "iter: " << iter << " , chi= " << currentChi_ << " , Lambda= " << currentLambda_
            << std::endl;
//------------------------------------------------ // new
// Lambda_data << "iter: " << iter << " , chi= " << currentChi_ << " , Lambda= " << currentLambda_<<endl;
 Lambda_data <<iter << "," << currentChi_ << " , " << currentLambda_<<endl;
//------------------------------------------------// end
```

- 通过将需要显示的值 导入在std::ofstream等定义的变量中，（方法很多，不一一列举）保存到对应格式文件中。 用python / matlab等绘制要求图像即可。
- **原始数据拟合可能效果较差，大家可以尝试调整数据，比如数据点个数，噪声大小（方差），增大数据范围，迭代次数等等，直到得到一个满意的拟合结果。**

第一题

T1.2： y函数改变 -> 对应改变残差函数ComputeResidual() -> 改变雅克比Jacobian于ComputeJacobians()函数中。

$$
\begin{aligned}
y &= ax_i^2 + bx_i + c \\
err_i &= ax_i^2 + bx_i + c \\
Jacobians_i &= [x_i^2, x_i, 1]
\end{aligned}
\tag{1}
$$

第一题

T1.3：策略1

$$\left[\mathbf{J}^\mathsf{T}\mathbf{W}\mathbf{J} + \lambda \boxed{\operatorname{diag}(\mathbf{J}^\mathsf{T}\mathbf{W}\mathbf{J})}\right] \mathbf{h}_{\mathsf{lm}} = \mathbf{J}^\mathsf{T}\mathbf{W}(\mathbf{y} - \hat{\mathbf{y}}) \ . \tag{13}$$

$$\rho_i(\mathbf{h}_{\mathsf{lm}}) = \frac{\chi^2(\mathbf{p}) - \chi^2(\mathbf{p}+\mathbf{h}_{\mathsf{lm}})}{(\mathbf{y}-\hat{\mathbf{y}})^\mathsf{T}(\mathbf{y}-\hat{\mathbf{y}}) - (\mathbf{y}-\hat{\mathbf{y}}-\mathbf{J}\mathbf{h}_{\mathsf{lm}})^\mathsf{T}(\mathbf{y}-\hat{\mathbf{y}}-\mathbf{J}\mathbf{h}_{\mathsf{lm}})} \tag{14}$$

$$= \frac{\chi^2(\mathbf{p}) - \chi^2(\mathbf{p}+\mathbf{h}_{\mathsf{lm}})}{\mathbf{h}_{\mathsf{lm}}^\mathsf{T}\left(\lambda_i \mathbf{h}_{\mathsf{lm}} + \mathbf{J}^\mathsf{T}\mathbf{W}\left(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{p})\right)\right)} \quad \boxed{\text{if using eq'n (12) for } \mathbf{h}_{\mathsf{lm}}} \tag{15}$$

$$= \frac{\chi^2(\mathbf{p}) - \chi^2(\mathbf{p}+\mathbf{h}_{\mathsf{lm}})}{\mathbf{h}_{\mathsf{lm}}^\mathsf{T}\left(\lambda_i \boxed{\operatorname{diag}(\mathbf{J}^\mathsf{T}\mathbf{W}\mathbf{J})} \mathbf{h}_{\mathsf{lm}} + \mathbf{J}^\mathsf{T}\mathbf{W}\left(\mathbf{y} - \hat{\mathbf{y}}(\mathbf{p})\right)\right)} \quad \boxed{\text{if using eq'n (13) for } \mathbf{h}_{\mathsf{lm}}} \tag{16}$$

1. $\lambda_0 = \lambda_o$; λ_o is user-specified [5].
   use eq'n (13) for $\mathbf{h}_{\mathsf{lm}}$ and eq'n (16) for ρ
   if $\rho_i(\mathbf{h}) > \epsilon_4$: $\mathbf{p} \leftarrow \mathbf{p} + \mathbf{h}$; $\lambda_{i+1} = \max[\lambda_i / L_\downarrow, 10^{-7}]$;
   otherwise: $\lambda_{i+1} = \min[\lambda_i L_\uparrow, 10^7]$;

- h_{lm}即δx；λ_i（$lambda$）即μ。此时注意δx和ρ的更新策略公式与前面不同。
- 需要更新对应的`Problem::AddLambdatoHessianLM()`和`Problem::RemoveLambdaHessianLM()`，以及`Problem::IsGoodStepInLM()`函数的实现方式。

第一题

T1.3：策略2

策略2和策略3的Δx和 ρ 更新是一样的。

$$\left[\mathbf{J}^\mathsf{T}\mathbf{W}\mathbf{J} + \lambda\mathbf{I}\right]\mathbf{h}_{\mathsf{lm}} = \mathbf{J}^\mathsf{T}\mathbf{W}(\mathbf{y}-\hat{\mathbf{y}}) \;, \qquad (12)$$

$$\rho_i(\mathbf{h}_{\mathsf{lm}}) = \frac{\chi^2(\mathbf{p}) - \chi^2(\mathbf{p}+\mathbf{h}_{\mathsf{lm}})}{(\mathbf{y}-\hat{\mathbf{y}})^\mathsf{T}(\mathbf{y}-\hat{\mathbf{y}}) - (\mathbf{y}-\hat{\mathbf{y}}-\mathbf{J}\mathbf{h}_{\mathsf{lm}})^\mathsf{T}(\mathbf{y}-\hat{\mathbf{y}}-\mathbf{J}\mathbf{h}_{\mathsf{lm}})} \qquad (14)$$

$$= \frac{\chi^2(\mathbf{p}) - \chi^2(\mathbf{p}+\mathbf{h}_{\mathsf{lm}})}{\mathbf{h}_{\mathsf{lm}}^\mathsf{T}\left(\lambda_i\mathbf{h}_{\mathsf{lm}} + \mathbf{J}^\mathsf{T}\mathbf{W}\left(\mathbf{y}-\hat{\mathbf{y}}(\mathbf{p})\right)\right)} \qquad \boxed{\text{if using eq'n (12) for } \mathbf{h}_{\mathsf{lm}} \text{ (15)}}$$

2. $\lambda_0 = \lambda_o \max\left[\mathrm{diag}[\mathbf{J}^\mathsf{T}\mathbf{W}\mathbf{J}]\right]$; λ_o is user-specified.
   use eq'n (12) for $\mathbf{h}_{\mathsf{lm}}$ and eq'n (15) for ρ
   $$\boxed{\alpha = \left(\left(\mathbf{J}^\mathsf{T}\mathbf{W}(\mathbf{y}-\hat{\mathbf{y}}(\mathbf{p}))\right)^\mathsf{T}\mathbf{h}\right) / \left(\left(\chi^2(\mathbf{p}+\mathbf{h}) - \chi^2(\mathbf{p})\right)/2 + 2\left(\mathbf{J}^\mathsf{T}\mathbf{W}(\mathbf{y}-\hat{\mathbf{y}}(\mathbf{p}))\right)^\mathsf{T}\mathbf{h}\right);}$$
   if $\rho_i(\alpha\mathbf{h}) > \epsilon_4$: $\mathbf{p} \leftarrow \mathbf{p} + \alpha\mathbf{h}$; $\lambda_{i+1} = \max\left[\lambda_i/(1+\alpha), 10^{-7}\right]$;
   otherwise: $\lambda_{i+1} = \lambda_i + |\chi^2(\mathbf{p}+\alpha\mathbf{h}) - \chi^2(\mathbf{p})|/(2\alpha)$;

第一题

T1.3：策略2 策略2相对难些，所以着重讲下程序的流程。

1. 计算ρ 更新需要的 α和δx

$$\alpha = \frac{((J^T W(y-\hat{y}(p)))^T h)}{(\frac{(\chi^2(p+h)-\chi^2(p))}{2} + 2(J^T W(y-y^\wedge(p)))^T h)} \tag{2}$$

2. 计算 $\delta x = \delta x * \alpha$ ， 并更新状态。

   δx 改变 $x = x + \delta x$ 中的 x 也发生改变，所以残差`tempChi` 即$\chi^2(p+h_{lm})$也要更新。

3. 计算`scale` (公式15的分母)，之后计算 ρ 即可

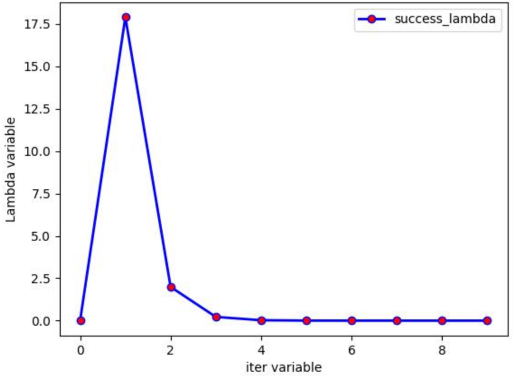
$$\rho = \frac{\chi^2(p) - \chi^2(p+h_{lm}))}{h_{lm}^T(\lambda_i h_{lm} + J^T W(y-\hat{y}(p))} \tag{3}$$

4. 按照策略2的描述，进行 λ 的相应更新即可。

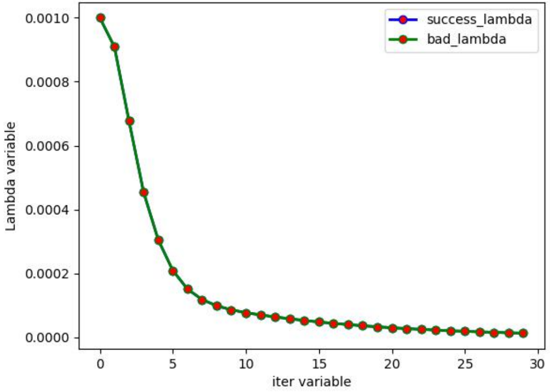
第一题

T1.3：策略3-即原代码中所用方案。最后，对比下不同策略，所输出阻尼因子变化(使误差下降的)

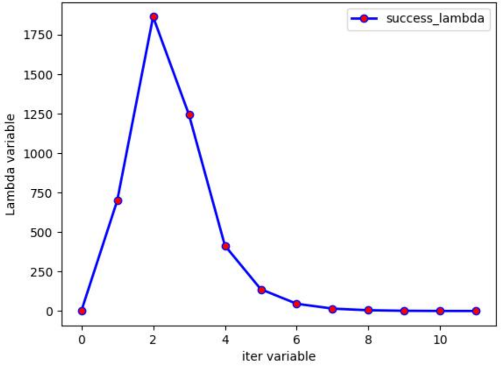
策略1



策略2



策略3



策略1：成功迭代次数 10/30
策略2：成功迭代次数 30/30
策略3：成功迭代次数 12/30

第二题

- T2:求解f15

已知:

$$\alpha_{b_k b_{k-1}} = \alpha_{b_i b_k} + \beta_{b_i b_k}\delta t + \frac{1}{2}a\delta t^2 \tag{4}$$

$$a = \frac{1}{2}(q_{b_i b_k}(a^{b_k} - b_k^a) + q_{b_i b_{k+1}}(a^{b_{k+1}} - b_k^a)) \tag{5}$$

$$= \frac{1}{2}(q_{b_i b_k}(a^{b_k} - b_k^a) + q_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}w\delta t \end{bmatrix}(a^{b_{k+1}} - b_k^a)) \tag{6}$$

$$w = \frac{1}{2}((w^{b_k} - b_k^g) + (w^{b_{k+1}} - b_k^g)) \tag{7}$$

$$= \frac{1}{2}(w^{b_k} + w^{b_{k+1}}) - b_k^g \tag{8}$$

$$f_{15} = \frac{\partial \alpha_{b_i b_{k+1}}}{\partial \delta b_k^g} = \frac{\partial \frac{1}{4} q_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(w - b_k^g)\delta t \end{bmatrix}(a^{b_{k+1}} - b_k^a)\delta t^2}{\partial b_k^g} \tag{9}$$

$$= \frac{\partial \frac{1}{4} R_{b_i b_k} exp([(w - b_k^g)\delta t]_\times)(a^{b_{k+1}} - b_k^a)\delta t^2}{\partial b_k^g} \tag{10}$$

$$= \frac{\partial \frac{1}{4} R_{b_i b_k} exp([w\delta t]_\times) exp([(-J_r(w\delta) b_k^g \delta t)]_\times)(a^{b_{k+1}} - b_k^a)\delta t^2}{\partial b_k^g} \tag{11}$$

$$= \frac{\partial \frac{1}{4} R_{b_i b_{k+1}}(I - [J_r(w\delta) b_k^g \delta t]_\times)(a^{b_{k+1}} - b_k^a)\delta t^2}{\partial b_k^g} \tag{12}$$

$$= \frac{\partial \frac{1}{4} R_{b_i b_{k+1}}[-J_r(w\delta) b_k^g \delta t]_\times (a^{b_{k+1}} - b_k^a)\delta t^2}{\partial b_k^g} \tag{13}$$

$$= -\frac{\partial \frac{1}{4} R_{b_i b_{k+1}}[(a^{b_{k+1}} - b_k^a)\delta t^2]_\times (-J_r(w\delta) b_k^g \delta t)}{\partial b_k^g} \tag{14}$$

$$= -\frac{1}{4} R_{b_i b_{k+1}}[(a^{b_{k+1}} - b_k^a)\delta t^2]_\times (-J_r(w\delta)\delta t) \tag{15}$$

$$= -\frac{1}{4} R_{b_i b_{k+1}}[(a^{b_{k+1}} - b_k^a)\delta t^2]_\times (-\delta t) \tag{16}$$

$$= \frac{1}{4} R_{b_i b_{k+1}}(a^{b_{k+1}} - b_k^a)^\wedge \delta t^3 \tag{17}$$

第二题

- T2：求解g12

已知：

$$\alpha_{b_k b_{k-1}} = \alpha_{b_i b_k} + \beta_{b_i b_k}\delta t + \frac{1}{2}a\delta t^2 \tag{18}$$

$$a = \frac{1}{2}(q_{b_i b_k}(a^{b_k} - b_k^a) + q_{b_i b_{k+1}}(a^{b_{k+1}} - b_k^a)) \tag{19}$$

$$= \frac{1}{2}(q_{b_i b_k}(a^{b_k} - b_k^a) + q_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}w\delta t \end{bmatrix}(a^{b_{k+1}} - b_k^a)) \tag{20}$$

$$w = \frac{1}{2}((w^{b_k} + n_k^g - b_k^g) + (w^{b_{k+1}} + n_{k+1}^g - b_k^g)) \tag{21}$$

$$= \frac{1}{2}(w^{b_k} + w^{b_{k+1}}) - b_k^g + \frac{1}{2}n_k^g + \frac{1}{2}n_{k+1}^g \tag{22}$$

$$g_{12} = \frac{\partial \frac{1}{4} q_{b_i b_k} \otimes \begin{bmatrix} 1 \\ \frac{1}{2}(w + \frac{1}{2}n_k^g)\delta t \end{bmatrix} (a^{b_{k+1}} - b_k^g)\delta t^2}{\partial n_k^g} \tag{23}$$

$$= \frac{\partial \frac{1}{4} R_{b_i b_k} exp([w\delta t]_\times) exp([J_r(w\delta t)\frac{1}{2}n_k^g \delta t]_\times)(a^{b_{k+1}} - b_k^g)\delta t^2}{\partial n_k^g} \tag{24}$$

$$= -\frac{\partial \frac{1}{4} R_{b_i b_{k+1}} [(a^{b_{k+1}} - b_k^g)\delta t^2]_\times J_r(w\delta t)\frac{1}{2}n_k^g \delta t}{\partial n_k^g} \tag{25}$$

$$= -\frac{1}{4} R_{b_i b_{k+1}} [(a^{b_{k+1}} - b_k^g)\delta t^2]_\times (\frac{1}{2}\delta t) \tag{26}$$

$$= -\frac{1}{8} R_{b_i b_{k+1}} (a^{b_{k+1}} - b_k^g)^\wedge \delta t^3 \tag{27}$$

第三题

- 证明：

$$(J^TJ+\mu I)\Delta x_{lm} = (V\Lambda V^T + \mu IVV^T)\Delta x_{lm} = V(\Lambda+\mu I)V^T\Delta x_{lm} \quad (28)$$

$$= -J^Tf = -F'^T \quad (29)$$

所以：$$\Delta x_{lm} = -\frac{V^TF'^T}{\Lambda+\mu I}V = -\sum_{j=1}^{n}\frac{v_j^TF'^T}{\lambda_j+\mu}v_j \quad (30)$$

所以：$$\Delta x_{lm} = -V(\Lambda+\mu I)^{-1}V^TF'^T \quad (31)$$

$$= -[v_1, v_2 \quad \cdots \quad v_n]\begin{bmatrix} \frac{1}{\lambda_1+\mu} & & & \\ & \frac{1}{\lambda_2+\mu} & & \\ & & \ddots & \\ & & & \frac{1}{\lambda_n+\mu}\end{bmatrix}\begin{bmatrix} v_1^T \\ v_2^T \\ \vdots \\ v_n^T\end{bmatrix}F'^T \quad (32)$$

$$= -[v_1, v_2 \quad \cdots \quad v_n]\begin{bmatrix} \frac{v_1^TF'^T}{\lambda_1+\mu} \\ \frac{v_2^TF'^T}{\lambda_2+\mu} \\ \vdots \\ \frac{v_n^TF'^T}{\lambda_n+\mu}\end{bmatrix} \quad (33)$$

$$= -\left(\frac{v_1^TF'^T}{\lambda_1+\mu}v_1 + \frac{v_2^TF'^T}{\lambda_2+\mu}v_2 + \cdots + \frac{v_n^TF'^T}{\lambda_n+\mu}v_n\right) = -\sum_{j=1}^{n}\frac{v_j^TF'^T}{\lambda_j+\mu}v_j \quad (34)$$

在线问答

Q&A

深蓝学院
shenlanxueyuan.com

感谢各位聆听！

Thanks for Listening