

## 非线性优化之高斯牛顿法、L-M 算法

2019-07-01 · SLAM · 约 2752 字 · 预计阅读 6 分钟

### 1. 状态估计中的最小二乘问题

#### 1.1 贝叶斯法则

- SLAM 要解决的两个问题是**定位（求解相机位姿）**和**建图（求解路标的位置）**，因此可以分别用一个**运动方程**和**观测方程**来建模 SLAM 问题：

$$\begin{cases} x_k = f(x_{k-1}, u_k) + w_k \\ z_{k,j} = h(y_i, x_k) + v_{k,j} \end{cases}$$

- 其中第一个公式称为**运动方程**， $x_k$  表示  $k$  时刻相机的位姿，其受到上一时刻  $k-1$  时刻的位姿  $x_{k-1}$  和当前时刻  $k$  传感器输入（比如 IMU 提供的旋转、相机计算出的  $R, t$ ）的影响。 $w_k$  是噪声项，满足 0 均值的高斯分布；
- 第二个公式称为**观测方程**， $z_{k,j}$  表示在相机位姿  $x_k$  时对路标点  $y_i$  产生的观测（图像上的像素位置）， $v_{k,j}$  是噪声项，满足 0 均值的高斯分布。
- 将所有待估计的量放入一个状态变量中  $x = \{x_1, \dots, x_N, y_1, \dots, y_N\}$ ，于是对机器人的状态估计就是已知输入数据  $u$  和观测数据  $z$  的情况下，求  $x$  的条件概率分布（这里的  $x, z, u$  都是数据的统称，不对应上面的两个方程）

$$P(x|z, u)$$

- 如果不考虑图像在连续时间上的运动问题，即没有传感器的运动测量，只有图像提供的观测，问题就变成了 **SFM**，变成估计  $P(x|z)$ 。
- 然后介绍几个概念
  - 先验**：指根据以往经验和分析得到当前的概率（在这里就是根据 0 ~  $k-1$  时刻的状态来推测  $k$  时刻的状态）；
  - 后验**：事件已经发生了，但发生可能有很多原因，判断事情发生时由哪个原因引起的概率（在这里可以理解为已知  $k$  时刻的状态，求这个状态是由观测  $z$  引起的概率）；
  - 似然**：事件已经发生，可能的原因有哪些（在现在的位姿下，可能产生怎样的观测数据）；
  - 最大似然**：在什么样的状态下，最可能产生现在的观测。
- 为估计状态变量的条件分布，利用贝叶斯法则有：

$$P(x|z) = \frac{P(z|x)P(x)}{P(z)} \propto P(z|x)P(x)$$

- 一个形象的例子（[参考](#)）

$$\begin{aligned}
 P(\text{加了醋}|\text{吃起来是酸的}) &= \frac{P(\text{加了醋, 且吃起来是酸的})}{P(\text{吃起来是酸的})} \\
 &= \frac{P(\text{吃起来是酸的}|\text{加了醋}) * P(\text{加了醋})}{P(\text{吃起来是酸的})} \\
 &= \frac{P(\text{吃起来是酸的}|\text{加了醋}) * P(\text{加了醋})}{P(\text{吃起来是酸的}|\text{加了醋}) * P(\text{加了醋}) + P(\text{吃起来是酸的}|\text{肉变质}) * P(\text{肉变质})}
 \end{aligned}$$

- 公式中左侧  $P(x|z)$  称为**后验概率**，右侧的  $P(z|x)$  称为**似然**， $P(x)$  称为**先验**；
- 直接求后验分布是困难的，但是求一个状态最优估计，使得该状态下**后验概率最大化**（MAP）是可行的（贝叶斯法则的分母部分与待估计的状态量  $x$  无关，因此忽略）

$$x^*_{MAP} = \operatorname{argmax} P(x|z) = \operatorname{argmax} P(z|x)P(x)$$

- 也就是说求解 **后验 = 最大似然 × 先验**，当没有相机位姿先验时，转变为求解  $x$  的**最大似然估计**：在怎样的状态  $x$  下，最可能产生现在的观测  $z$

$$x^*_{MLE} = \operatorname{argmax} P(z|x)$$

## 1.2 最小二乘的引出

- 在高斯分布下，最大似然有较简洁的形式，可以通过最小化负对数的方式来求解一个高斯分布的最大似然。**
- 回顾前面的**观测方程**  $z_{k,j} = h(y_i, x_k) + v_{k,j}$ ，假设其噪声项服从 0 均值的高斯分布  $v_k \sim N(0, Q_{k,j})$ ，则其观测数据的条件概率为：

$$P(z_{j,k}|x_k, y_j) \sim N(h(y_i, h_k), Q_{k,j})$$

- 考虑任意一个高维高斯分布  $x \sim N(\mu, \Sigma)$ ，其**概率密度函数**展开的形式为：

$$P(x) = \frac{1}{\sqrt{(2\pi)^N \det(\Sigma)}} \exp\left(-\frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu)\right)$$

- 对上式**取负对数**（高斯分布在负对数下有较好的数学形式）有

$$-\ln(P(x)) = \frac{1}{2} \ln\left((2\pi)^N \det(\Sigma)\right) + \frac{1}{2}(x - \mu)^T \Sigma^{-1} (x - \mu).$$

- 对原分布求最大化相当于**对负对数求最小化**，在最小化上式的  $x$  时，第一项与  $x$  无关，可以略去。于是，**只要最小化右侧的二次型项，就得到了对状态的最大似然估计。**
- 将上式带入到 SLAM 模型中，得到

$$x^* = \arg \min \left( (z_{k,j} - h(x_k, y_j))^T Q_{k,j}^{-1} (z_{k,j} - h(x_k, y_j)) \right)$$

- 观察上式会发现相当于**最小化噪声项（即误差项）的平方**
- 对于所有的运动和任意的观测，定义**数据与观测值之间的误差**为：

$$e_{v,k} = x_k - f(x_{k-1}, u_k)$$

$$e_{y,j,k} = z_{k,j} - h(x_k, y_j)$$

- 从而求得该**误差的平方之和**

$$J(x) = \sum_k e_{v,k}^T R_k^{-1} e_{v,k} + \sum_k \sum_j e_{y,k,j}^T Q_{k,j}^{-1} e_{y,k,j}$$

- 上式即为 SLAM 的最小二乘问题，通过不断调整误差量使之达到一个极小值，从而得到最准确的状态估计。

## 2. 最小二乘问题的一阶二阶梯度法

- 对于方便直接求解的函数，通过使目标函数的导数为 0，然后逐个比较导数为 0 处的极值；对于不方便直接求解的最小二乘问题，使用迭代的方式，从一个初始值出发，不断地更新当前的优化变量，使目标函数下降，重点是要寻找到每次增量  $\Delta x$ 。

### 2.1 最速下降法

- 求解增量最直接的方式是将目标函数在  $x$  附近进行泰勒展开

$$\|f(x + \Delta x)\|_2^2 \approx \|f(x)\|_2^2 + J(x) \Delta x + \frac{1}{2} \Delta x^T H \Delta x$$

- 其中  $J$  是关于  $x$  的导数（雅克比矩阵）， $H$  是二阶导（海森矩阵），可以选择对其保留一阶项或二阶项。
- 如果只保留一阶项，则增量的方向为：

$$\Delta x^* = -J^T(x)$$

- 其意义为沿着逆梯度的方向进行迭代（还需要取一个步长），求得最快的下降方式，称为最速下降法，其缺点是过于贪心，容易走出锯齿路线，反而增加了迭代次数。

### 2.2 牛顿法

- 如果对上面的泰勒展开式保留至二阶项，则增量的方程为：

$$\Delta x^* = \arg \min \|f(x)\|_2^2 + J(x) \Delta x + \frac{1}{2} \Delta x^T H \Delta x$$

- 求右侧等式关于  $\Delta x$  的导数，并令其为 0，得到增量的解

$$H \Delta x = -J^T \Rightarrow \Delta x = -H^{-1} J^T$$

- 这个方法称为牛顿法，二阶收敛，但需要计算目标函数的  $H$  矩阵，这在问题规模较大时非常困难，我们通常倾向于避免  $H$  的计算

## 3. 高斯-牛顿法

- 对于一个非线性最小二乘问题  $x = \arg \min \frac{1}{2} \|f(x)\|^2$ ，高斯牛顿法的思想是对  $f(x)$ （而不是目标函数  $f(x)^2$ ）进行泰勒展开，取一阶线性项近似

$$f(x + \Delta x) \approx f(x) + J(x) \Delta x$$

- 将其带入到最小二乘问题中，得到

$$\begin{aligned}\frac{1}{2}\|f(x) + J(x)\Delta x\|^2 &= \frac{1}{2}(f(x) + J(x)\Delta x)^T(f(x) + J(x)\Delta x) \\ &= \frac{1}{2}\left(\|f(x)\|_2^2 + 2f(x)^T J(x)\Delta x + \Delta x^T J(x)^T J(x)\Delta x\right).\end{aligned}$$

◦ 对上式关于  $\Delta x$  求导，令导数为 0 得到

$$J(x)^T J(x)\Delta x = -J(x)^T f(x)$$

- 上式即为**高斯牛顿方程**（增量方程、正规方程），要求的变量是  $\Delta x$ ，是一个线性方程，将左侧的系数定义为  $H$ ，右侧定义为  $g$ ，上式变成

$$H\Delta x = g$$

◦ 注意上式左侧的  $H$ ，**高斯牛顿法中用  $J^T J$  来近似牛顿法中的二阶海森矩阵，从而省略了  $H$  矩阵的求解过程（所以高斯-牛顿是近似二阶泰勒展开）**。

#### • 迭代步骤

1. 给定初始值  $x_0$ 。
2. 对于第  $k$  次迭代，求出当前的雅可比矩阵  $J(x_k)$  和误差  $f(x_k)$ 。
3. 求解增量方程： $H\Delta x_k = g$ 。
4. 若  $\Delta x_k$  足够小，则停止。否则，令  $x_{k+1} = x_k + \Delta x_k$ ，返回 2。

#### • 缺陷

- 高斯牛顿要求所用的近似  $H$  矩阵是可逆的(而且是正定的)，但实际数据中计算得到的  $J^T J$  只有半正定性，也就是说,在使用 Gauss Newton 方法时,可能出现  $J^T J$  为奇异矩阵或者病态的情况，此时增量的稳定性较差，导致算法不收敛；
- 更严重的是,就算我们假设  $H$  非奇异也非病态，如果求出来的步长  $\Delta x$  太大，也会导致我们采用的局部近似(一阶泰勒展开式)不够准确，这样一来我们甚至都无法保证它的迭代收敛，哪怕是让目标函数变得更大都是有可能的。

## 4. L-M 法

- G-N 方法中采用的近似二阶泰勒展开只能在展开点附近有较好的近似结果，**在 L-M 算法中给增量添加了一个信任域**，这个区域由近似模型和实际函数之间的差异来确定，如果差异小，可以让范围尽量大；如果差异大，则缩小近似范围。
- L-M 算法的优化问题：

$$\min_{\Delta x_k} \frac{1}{2}\|f(x_k) + J(x_k)\Delta x_k\|^2, \quad s.t. \|D\Delta x_k\|^2 \leq \mu$$

- 其中  $\mu$  是信任区域的半径，将增量限定在一个半径为  $\mu$  的球内，约束项带上  $D$  之后，这个球可以看成是一个椭球；
  - 在 Levenberg 提出的优化方法中，将  $D$  取单位矩阵  $I$ ，相当于直接把  $\Delta x$  约束在一个球中；
  - 之后 Marquardt 剔除将  $D$  取成非负数对角阵（实际中通常取  $J^T J$  的对角元素平方根），使得在**梯度小的维度上约束范围更大一点**。
- 利用拉格朗日乘子将上面的有约束优化问题转换成无约束优化问题

$$\min_{\Delta \mathbf{x}_k} \frac{1}{2} \|f(\mathbf{x}_k) + \mathbf{J}(\mathbf{x}_k) \Delta \mathbf{x}_k\|^2 + \frac{\lambda}{2} \|D \Delta \mathbf{x}\|^2$$

- 其中  $\lambda$  是拉格朗日乘子。类似于前面高斯牛顿的方法，将其展开，其实该问题的核心仍然是计算增量的线性方程

$$(H + \lambda D^T D) \Delta \mathbf{x} = g$$

- 上式增量方程相比于 G-N 多了意向  $\lambda D^T D$ ，先将其带入简化的形式，即  $D = I$ ，相当于是求解

$$(H + \lambda I) \Delta \mathbf{x} = g$$

- 对于上式增量方程可以发现：
  - 当参数  $\lambda$  较小时， $H$  占主导地位，说明二次近似模型在该范围内比较好，**L-M 算法更接近于 G-N 算法**；
  - 当参数  $\lambda$  较大时， $\lambda I$  占主导，L-M 算法更接近于一阶梯度下降法（最速下降法）。

## R. 参考资料

- [1] SLAM 十四讲第 6 讲
- [2] 贝叶斯公式的直观理解(先验概率/后验概率)
- [3] [优化] Gauss-Newton非线性最小二乘算法
- [4] Gauss-Newton算法学习 (☆)

wuyanminmax@gmail.com  
2019.07.01

< 😊 ORB-SLAM2 代码解读 (三) : 优化 1 (概述)

😊 ORB-SLAM2 代码解读 (三) : 单目初始化 >



© 2019 - 2020 ♥ wu