



General Graph Optimization

在 g2o 中选择优化方法一共需要三个步骤：

1. 选择一个线性方程求解器: PCG, CSparse, Choldmod;
2. 选择一个 BlockSolver。 , 一个 SparseBlockMatrix，用于计算稀疏雅可比和海塞；一个是用于计算迭代过程中最关键的一步 $H \Delta x = -b$ 为一个线性方程的求解器。
3. 选择一个迭代策略: GN, LM, Doglog。

图是由顶点（Vertex）和边（Edge）组成的结构，而图论则是研究图的理论。我们记一个图为 $G=\{V,E\}$ ，其中 V 为顶点集， E 为边集。

顶点没什么可说的，想象成普通的点即可。边是什么呢？一条边连接着若干个顶点，表示顶点之间的一种关系。边可以是有向的或是无向的，对应的图称为有向图或无向图。边也可以连接一个顶点（Unary Edge，一元边）、两个顶点（Binary Edge，二元边）或多个顶点（Hyper Edge，多元边）。最常见的边连接两个顶点。当一个图中存在连接两个以上顶点的边时，称这个图为超图（Hyper Graph）。

图优化的流程：

- 1.选择你想要的图里的节点与边的类型，确定它们的参数化形式；
- 2.往图里加入实际的节点和边；
- 3.选择初值，开始迭代；
- 4.每一步迭代中，计算对应于当前估计值的雅可比矩阵和海塞矩阵；
- 5.求解稀疏线性方程 $H_k \Delta x = -b_k$ ，得到梯度方向；
- 6.继续用 GN 或 LM 进行迭代。如果迭代结束，返回优化值。

实际上，g2o 能帮你做好第 3-6 步，你要做的只是前两步而已。

核函数

引入核函数的原因，是因为 SLAM 中可能给出错误的边。SLAM 中的数据关联让科学家头疼了很长时间。出于变化、噪声等原因，机器人并不能确定它看到的某个路标，就一定是数据库中的某个路标。万一认错了呢？把一条原本不应该加到图中的边给加进去了，会怎么样？那优化算法可就懵逼了，它会看到一条误差很大的边，然后试图调整这条边所连接的节点的估计值，使它们顺应这条边的无理要求。由于这个边的误差真的很大，往往会抹平了其他正确边的影响，使优化算法专注于调整一个错误的值。

于是就有了核函数的存在。核函数保证每条边的误差不会大的没边，掩盖掉其他的边。具体的方式是，把原先误差的二范数度量，替换成一个增长没有那么快的函数，同时保证自己的光滑性质（不然没法求导）。因为它们使得整个优化结果更为鲁棒，所以又叫它们为 robust kernel（鲁棒核函数）。

很多鲁棒核函数都是分段函数，在输入较大时给出线性的增长速率，例如 cauchy 核，huber 核等等。

核函数在许多优化环境中都有应用，有一大堆人在机器学习算法里加各种各样的核，现在常用的 svm 也会带个核函数。