

STAT243 Lecture 10.7 Iterative solutions of linear systems (optional)

1 Gauss–Seidel

我们希望通过迭代方法求解线性系统 $Ax = b$ 。Gauss–Seidel 方法逐个更新 x 的分量，算法如下：

- 从一个初值 $x^{(0)}$ 开始
- 固定除 $x_1^{(0)}$ 以外的所有分量，通过

$$x_1^{(1)} = \frac{1}{a_{11}} \left(b_1 - \sum_{j=2}^n a_{1j} x_j^{(0)} \right)$$

得到新的 $x_1^{(1)}$

- 对 A 的每一行重复上述步骤，得到 $x^{(1)}$
- 继续得到 $x^{(2)}, x^{(3)}, \dots$

典型收敛判据包括：

- $|x^{(k)} - x^{(k-1)}| \leq \epsilon$
- 或 $|r^{(k)} - r^{(k-1)}| \leq \epsilon$, 其中 $r^{(k)} = b - Ax^{(k)}$

每次更新单个 x_i 的代价为 $O(n)$ ，因此一次完整迭代的代价为 $O(n^2)$ 。若能在少于 n 次迭代收敛，则比直接求解（如 LU 分解）更快。

若将矩阵分解为

$$A = L + D + U$$

其中 L 为严格下三角， U 为严格上三角，则 Gauss–Seidel 等价于求解

$$(L + D)x^{(k+1)} = b - Ux^{(k)}$$

而求解下三角系统的代价仍是 $O(n^2)$ 。

收敛速度取决于矩阵

$$(L + D)^{-1}U$$

的谱半径。几何意义上，Gauss–Seidel 相当于沿坐标方向优化，因此某些情况下收敛较慢。

2 Conjugate gradient (CG)

当矩阵 A 为 对称正定 时，CG 是求解 $Ax = b$ 的最常用迭代方法。其思路是将问题视为最小化以下二次函数：

$$f(x) = \frac{1}{2}x^\top Ax - x^\top b$$

其梯度为 $Ax - b$ ，因此最小点即是线性系统的解。

与收敛缓慢的最速下降法不同，CG 构造一组关于 A 共轭的方向 d_k ，并沿这些方向前进：

$$x_{(k)} = x_{(k-1)} + \alpha_k d_k$$

基本算法：

- 给定初始值 $x_{(0)}$
- 定义残差 $r_{(0)} = b - Ax_{(0)}$
- 设方向 $d_0 = r_{(0)}$, 令 $k = 0$
- 迭代以下步骤：

- 步长：

$$\alpha_k = \frac{r_{(k)}^\top r_{(k)}}{d_k^\top A d_k}$$

- 更新解：

$$x_{(k+1)} = x_{(k)} + \alpha_k d_k$$

- 更新残差：

$$r_{(k+1)} = r_{(k)} - \alpha_k A d_k$$

- 更新方向：

$$d_{k+1} = r_{(k+1)} + \frac{r_{(k+1)}^\top r_{(k+1)}}{r_{(k)}^\top r_{(k)}} d_k$$

- 当 $|r_{(k+1)}|$ 小于某阈值时停止

CG 的性质：

- 理论上在 n 步内 (n 为维度) 可得到精确解
- 在实际计算中，由于数值误差，会提前终止
- 收敛速度取决于 A 的条件数
- 适用于大规模稀疏正定系统，是实际工程中最重要的方法之一

更详细资料可参考 [An Introduction to the Conjugate Gradient Method Without the Agonizing Pain](#)。

3 Updating a solution

若已求解 $Ax = b$ ，之后需要求解 $Ax = c$ ，可复用原有分解：

- 若已做 LU 或 Cholesky 分解，则只需 $O(n^2)$ 进行一次 forward/back substitution
- 若 $c = b + \delta b$ 且变化不大，可将旧解 x 作为迭代初值并使用 Gauss–Seidel 或 CG 快速收敛

在大型问题中，**重复使用 factorization 比重新分解矩阵能节省大量计算。