

共轭梯度法在大规模线性方程组中的应用

向平 21960197

摘要

本文分别使用基于对称连续过松弛和不完全Cholesky分解的预处理算子，结合共轭梯度法求解一个大规模稀疏矩阵线性方程组问题。文章描述了带预处理共轭梯度法的通用算法框架，并对不同的预处理算子，以及同一预处理算子在不同参数条件下的算法迭代效率进行对比。文章最后讨论了在该大规模线性方程组预处理算子的选择思路。

背景介绍

线性共轭梯度法求解如下优化问题：

\$\$

$$\min \phi(x) = \frac{1}{2} x^T A x - b^T x,$$

\$\$

$$\min \phi(x) \stackrel{\text{def}}{=} \frac{1}{2} x^T A x - b^T x, \quad (1)$$

要求矩阵 A 对称正定，该优化问题最优解满足 $\nabla \phi(x) = Ax - b \stackrel{\text{def}}{=} r(x) = 0$ 。自然，可以定义迭代过程中第 k 步中的残差为 $r_k = Ax_k - b$ 。给定一组共轭向量，即向量组 $\{p_0, p_1, \dots, p_{n-1}\}$ 满足 $p_i^T A p_j = 0, \forall i \neq j$ ，共轭方向法依次沿各共轭向量对 $\phi(x)$ 的最小值进行精确搜索，即

$$x_{k+1} = x_k + \alpha_k p_k, \quad (2)$$

其中 α_k 是最优步长

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}. \quad (3)$$

给定共轭方向集合 $\{p_0, p_1, \dots, p_{n-1}\}$ ，对于任意初始点 x_0 ，共轭方向算法最多 n 步即可收敛到最优值 x^* 。

共轭梯度算法是共轭方向算的一种，区别在于CG法提供了计算共轭方向集合的方法。第 k 步迭代的共轭方向 p_k 选择为当前残差方向和上一步共轭方向的线性表示，即 $p_k = -r_k + \beta_k p_{k-1}$ ，由于需要满足迭代方向共轭，即 $p_k^T A p_{k-1} = 0$ ，可以推出：

$$\beta_k = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}. \quad (4)$$

初始点 x_0 可以随机选择，初始搜索方向 p_0 选择为最速下降方向 $-r_0$ 。将该共轭方向计算步骤与共轭方向法结合即可得到最基本的共轭梯度法，但为了具体编程实现，还可以进一步优化。

定理 1：共轭梯度算法产生的解序列 $\{x_k\}$ 满足以下性质：

$$r_k^T r_i = 0, i = 0, 1, \dots, k-1, \quad (5)$$

$$p_k^T A p_i = 0, i = 0, 1, \dots, k-1. \quad (6)$$

即当前迭代步骤产生的残差与此前所有残差正交；并且当前迭代方向与此前所有方向共轭。基于定理1可以对上述基础共轭梯度法式（3）和式（4）进行优化，推理得到：

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}, \quad (7)$$

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}. \quad (8)$$

尽管共轭梯度法能保证 n 步收敛速度，但面对大规模稀疏矩阵的求解，我们还是希望借助稀疏矩阵的一些性质进行预处理操作，加快收敛速度。迭代算法的收敛率通常直接或间接依赖于系数矩阵 A 的特征值分布，或者说条件数，因此预条件方法通过降低问题的条件数以及调整系数矩阵特征值分布来加快收敛速度[1] [2]。

线性方程组的预条件形式如下：

$$M^{-1} A x = M^{-1} b. \quad (9)$$

如果取预条件矩阵 $M = A$ ，则问题的条件数为1，但求逆矩阵计算代价加大；为了方便求逆，对角占优的矩阵是最便利的选择，但单位矩阵 I 并不能降低条件数，因此 M 的选择可以取两者之间。最简单的方式是雅克比预条件子 $M = \text{diag}(A)$ 。更复杂的预条件子包括对称连续过松弛（SSOR），预条件子定义如下：

$$M = (D + \omega L) D^{-1} (D + \omega U), \quad (10)$$

其中 D 为 A 对角矩阵， L 为下三角矩阵， U 为上三角矩阵， ω 是 $(0, 2)$ 之间的常数，当 $\omega = 1$ 是为高斯-赛德尔预条件子。

此外常用的预处理算法还有基于不完全Cholesky分解，如下：

$$A = LL^T - R, \quad (11)$$

或者基于修正的不完全Cholesky分解预处理，如下：

$$Ae = LL^T e \quad (12)$$

则预处理子 $M = LL^T$ [3]。

算法设计

由前讨论可知，共轭梯度法基本步骤分为三步：（1）计算最优步长 α_k ；（2）计算共轭方向 p_k ；（3）计算 $x_{k+1} = x_k + \alpha_k p_k$ 。结合预处理算子 M ，CG法基本框架如下[1]：

预条件共轭梯度算法

给定 x_0 ，预条件算子 M ；

$r_0 \leftarrow b - Ax_0$ ；

Solve $My_0 = r_0$ ；

Set $p_0 = -y_0, k \leftarrow 0$ ；

while $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T y_k}{p_k^T A p_k}$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k$$

Solve $M y_{k+1} = r_{k+1}$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k$$

$$k \leftarrow k + 1$$

end (while)

仿真结果

总结

参考文献

- [1] Nocedal J, Wright S. Numerical Optimization, 2nd Edition[M]. Springer Science & Business Media, 2006: 118-120.
- [2] Sauer T. Numerical Analysis, 2nd Edition[M]. Pearson Addison Wesley, 2012:126-128.
- [3] Yousef S. Iterative Methods for Sparse Linear Systems, 2nd Edition[M]. PWS Publishing Company, 2000: 269-286.

求解大规模线性方程组在工业生产中有重要作用，共轭梯度（Conjugate Gradient, CG）法作为一种迭代算法，为稀疏矩阵问题的求解提供了新的解决方案。由于共轭梯度法的迭代效率与系数矩阵的特征分布直接相关[1]，或者说迭代效率依赖于系数矩阵的条件数[2]，因此通过适当的预处理方法调整系数矩阵的特征分布，能够大大加快算法的收敛效率。