ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.6,No.14, May 2010, pp.3720-3721

E-mail: eduf@cccc.net.cn
http://www.dnzs.net.cn
Tel:+86-551-5690963 5690964

动态规划方法求解矩阵连乘问题

彭东海

(湖南人文科技学院 计算机系,湖南 娄底 417000)

摘要:动态规划法是求解最优化问题的一种方法,该文主要研究其求解问题的基本思想及具体步骤,详细分析其用于矩阵链乘问题上的算法设计,并给出其算法实现。

关键词:动态规划;矩阵链乘问题;最优化原理

中图分类号:TP301 文献标识码:A 文章编号:文章编号:1009-3044(2010)14-3720-02

1 动态规划的思想

动态规划是一种将复杂的问题分解为更小的、相似的子问题,并存储子问题的解而避免计算重复的子问题,以解决最优化问题的算法策略。

动态规划主要应用于最优化问题,这类问题会有多种可能的解,每个解都有一个值,而动态规划找出其中最优解。在求解的过程中,动态规划也是通过求解局部子问题的解达到全局最优解,允许这些子问题不独立,也允许其通过自身子问题的解作出选择,对每一个子问题只解一次,而且把结果保存起来,避免重复计算来提高效率。

2 动态规划方法求解矩阵连乘问题

2.1 矩阵连乘问题

给定 n 个矩阵{D1,D2,…,Dn},其中 Di 与 Di+1 是可乘的,i=1,2,…,n-1。因为矩阵乘法满足结合律,所以计算 n 个矩阵的连乘积会有多种不同的计算顺序,而且计算顺序可以用加括号的方式来确定。假设 n 个矩阵连乘积的计算顺序确定,则可以按照顺序调用 2 个矩阵相乘的方法计算出 n 个矩阵的连乘积。若 D 是一个 r×s 矩阵,E 是一个 s×t 矩阵,计算乘积 C=DE 的方法中,需要进行 rst 次数乘法运算。

矩阵连乘积的计算顺序不同,总的也不同,举例如下:

D1,D2,D3 分别是 5*50,50*100 和 100*10 的矩阵。如果按照 ((D1D2)D3) 来计算, 则计算所需的总数乘次数是 5*50*100+5*100*10=30000。如果按照(D1(D2D3))来计算,则需要的数乘次数是 50*100*10+5*50*10=525000,整整是前者的 1.75 倍。因而在计算矩阵连乘积时,不同的加括号方式所导致的不同的计算对总计算量有很大的影响。如何确定计算 n 个矩阵连乘积的一个计算顺序,使得按照这个顺序计算矩阵连乘积所需要的数乘次数最少便成为一个的问题。

2.2 最优解的结构

把矩阵连乘积 DiDi+1...Dj 简记为 D[i:j]。对于 D[1:n]的一个最优顺序,设这个计算顺序在矩阵 Dk 和 Dk+1 之间将矩阵链断开(1<=k<n),那么完全加括号的方式为((D1...Dk)(Dk+1...Dn))。依此顺序,先分别计算 D[1:k]和 D[k+1:n],然后把计算结果相乘得到 D[1:n],总计算量为 D[1:k]的计算量加上 D[k+1:n]的计算量,再加上 D[1:k]和 D[k+1:n]相乘的计算量。

可以证明,问题的关键在于:计算 D[1:n]的一个最优顺序所包含的计算矩阵子链 D[1:k]和 D[k+1:n]的顺序也是最优的。因而矩阵连乘积计算顺序问题的最优解包含着其自身子问题的最优解。这种最优子结构性质是问题可以用动态规划方法来求解的最重要特征。

2.3 利用递归来定义最优值

假设计算 D[i:j](1<=i<=j<=n)所需最少的数乘次数为 a[i][j],则原问题的最优值为 a[1][n]。显然,当 i=j 时,a[i][j]=0。

根据上述最优子结构性质,当 i<j 时,若计算 D[i:j]的最优顺序在 Dk 和 Dk+1 之间断开,可以定义 a[i][j]=a[i][k]+a[k+1][j]+qi-1*qk*qj(其中,Di 的维数为 qi-1*qi)。因而有:

当 i=j 时,a[i][j]=0。

当 i<j 时,a[i][j]=min{a[i][k]+a[k+1][j]+qi-1* qk*qj} (i<=k<j)。

另外,若将对应于 a[i][j]的断开位置记为 r[i][j],计算出最优值 a[i][j]后,就可以递归地由 r[i][j]构造出相应的最优解。

2.4 计算最优值

如果直接用 a[i][j]的计算公式,进行递归计算需要耗费指数计算时间。然而不同的子问题的个数只是 n 的平方项级。用动态规划方法求解,可按照其递归式以自底向上的方式来计算。在计算过程中,保存已解决的子问题解。因而每个子问题只计算一次,在后面需要时只要查找一下,通过这种方法避免了大量的重复计算,因而得到多项式时间的算法。下面给出计算 a[i][j]的动态规划算法:

收稿日期:2010-03-16
基金项目:湖南省教育厅科学研究项目(08C457);湖南省重点学科建设基金资助项目
作者简介:彭东海(1975-),男,硕士,讲师,主要研究领域为算法设计、生物信息学。

```
void multiMatrix (int * q, int n, int * * a, int * * r){
    for ( int i=1;i<=n;i++)
        a[i][i]=0;
    for ( int f=2;f<=n;f++)
        for ( int i=1;i<=n-f+1;i++) {
            int j=i+f-1;
            a[i][j]=a[i+1][j]+q[i-1]*q[i]*q[j];
            r[i][j]=i;
            for(int k=i+1;k<j;k++) {
          int t=a[i][k]+a[k+1][j]+q[i-1]* q[k]*q[j];
                if (t<a[i][j]){
                    a[i][j]=t;
                    r[i][j]=k;
                } } } }
```

算法开始设置 a[i][i]=0(i=1,2,...,n),然后根据递归式按照矩阵链长的递增方式计算出各个 a[i][j],在计算某个固定的 a[i][j]时,只用到已经计算出的 a[i][k]和 a[k+1][j]。经过分析可以得出,算法用 O(n^2)的空间消耗大大降低了时间复杂度,算法时间的上界为 O(n^3)。

2.5 构造最优解

通过上面算法得到了计算所给矩阵连乘积所需的最少数乘次数,但是还不知道应该按照什么顺序来做两矩阵的简单乘法才能达到这个最少数乘次数。在 r[i][j]已经保存了构造最优解所需要的信息,从 r[1][n]记录的信息可以知道计算 D[1:n]的最优加括号方式为(D[1:r[1][n]])(D[r[1][n] +1:n])。同理,前后部分的最优加括号方式又可以根据数组 r 的相应元素得出。如此递推,最终可以确定 D[1:n]的最优完全加括号方式,即得出问题的一个最优解。

3 结束语

动态规划方法中,每步所作的选择往往依赖于相关子问题的解。因而只有在解出相关子问题后,才能做出选择,所以动态规划算法通常是以自底向上的方式解各子问题的解,进而求出原问题的解。 实际求解过程中,子问题会有大量的重复,而动态规划就是对重复出现的子问题,只在第一次遇到时进行求解,并把解保存起来,让以后再遇到时直接引用,不必重新求解。

参考文献:

[1] 吴中福.算法设计与分析[M].重庆:重庆大学出版社,2001.
[2] 王晓东.计算机算法设计与分析[M].北京:电子工业出版社,2004.
[3] 方世昌.算法设计技巧与分析[M].北京:电子工业出版社,2004.