NANJING UNIVERSITY

算法设计与分析基础《Introduction to the Design and Analysis of Algorithms》

动态规划

南京大学软件学院

李传艺

lcy@nju.edu.cn

费彝民楼917

简介

- 动态规划 Dynamic Programming
  - 一种“使多阶段决策过程最优”的通用方法
  - 一种算法设计技术：时空权衡思想

- 应用场景
  - 如果问题由交叠的子问题组成，并且能够给出子问题的解与给定问题的解之间的递推关系，将子问题逐步分解为更小的子问题，就可以使用动态规划方法、

- 关键：递推关系 => 递归？
  - 区别
    - 递归是保存求解过程中每一个步骤的计算空间，达到停止条件后，逐步退回——自顶向下
    - 动态规划是想直接从停止条件开始往要求解的结果计算，保存的只有中间结果——自底向上
  - 共同点
    - 找到递推关系

问题一：最长公共子序列LCS

- ADBCDACBA
- ABCADBACB
- 最长公共子序列：ABCDACB

- 蛮力解法
  - 对某一个序列(长度为m)的所有子串判断是否是另一个序列(长度n)的子串
  - 时间复杂度？
    - 每一次检查是O(n)
    - 共有2^m个子串
    - O(n*2^m)

- 递归解法
  - 递推关系

$$num[i][j]=\begin{cases}0 & \mathrm{i=0或者j=0}\\ 1+num[i-1][j-1] & \mathrm{i,j>0,a[i]=b[j]}\\ \max\{num[i][j-1],num[i-1][j]\} & \mathrm{i,j>0,a[i]\neq b[j]}\end{cases}$$

  - 递归调用树

问题一：最长公共子序列LCS

- 动态规划解法
  - 自底向上求解，使用额外空间记录中间结果
  - 计算num[i,j]从计算num[1,1]开始，保存num[i,j] (0<i<m, 0<j<n)所有值

$$num[i][j]=\begin{cases}0 & \mathrm{i}=0\text{或者}\mathrm{j}=0\\ 1+num[i-1][j-1] & \mathrm{i,j}>0,\mathrm{a[i]}=\mathrm{b[j]}\\ \max\{num[i][j-1],num[i-1][j]\} & \mathrm{i,j}>0,\mathrm{a[i]}\neq\mathrm{b[j]}\end{cases}$$

i j	1	2	3	4	5	6	…	m
1								
2								
3								
4								
5								
…								
n								

问题二：计算二次项系数

- (a+b)^n = C(n,0)a^n + C(n,1)a^(n-1)b^1 + … + C(n,k)a^(n-k)b^k + … + C(n,n)b^n

- 当n>k>0时，C(n,k) = C(n-1,k-1) + C(n-1,k)
- 当k=0或k=n时，C(n,0) = C(n,n) = 1

- 直接计算
  - C(n,k) = n!/k!/(n-k)!

- 递归解法
  - 递归调用树

- 动态规划
  - 计算C(n,k)时需要的加法次数？
  - 需要的额外空间？

问题三：计算最短路径

- 国际象棋中车可以水平或竖直一定到同行或同列的任意位置，那么一个车要从棋盘的一个角移动到对角，有多少种最短路径？
  - 设棋盘有n*n的格子
  - 什么是最短路径？：不回头
  - 初等排列组合求解？
    - 在2n个步骤中选出n个用来横向走，或者
    - 在2n个步骤中选出n个用来纵向走
    - C(2n,n)
  - 动态规划求解？设A(r,c)表示横向走了r步，纵向走了c步
    - A(r,c) = A(r-1,c) + A(r,c-1)，r、c>0
    - r=c=0时 A(0,0) = 0
    - r=0或c=0，A(r,0) = 1， A(0,c) = 1

i j	0	1	2	3	4	5	…	n
0	0	1	1	1	1	1	…	1
1	1							
2	1							
3	1							
4	1							
…	…							
n	1							

问题四：胜率问题

- 考虑A和B两支队伍，正在进行一系列比赛，直到一个队赢了n场，比赛才停止。A赢得每一场的概率都是p,而输掉的概率是q=1-p（表示没有平局）。当A队还需要赢i场才能赢得系列赛，B队还需要j场胜利才赢得系列赛的时候，A队赢得系列赛的概率是P(i,j)，请求解P(i,j)
  - P(i,j)的递推关系公式是什么？
  - 边界是什么？
    - P(0,0) or P(n,n)
  - 假设n=6，求P(6,6), P(6,5), P(5,6), P(5,5)?
  - $P(i,j) = p*P(i+1, j) + q*P(i, j+1)$
  - 课后完成算法的编程实现，计算n=10, p=0.3时P(3,6)的值

问题五：背包问题

- 给定n个重量为$w_1,w_2,\ldots,w_n$，价值为$v_1,v_2,\ldots,v_n$的物品和一个承重为W的背包，求这些物品中最有价值的一个子集，并且要能够装到背包中。

- 穷举法

- 动态规划方法
  - 用V[i,j]表示前i个物品中能够放入承重为j的背包的物品总价值，物品重量为$w_1,w_2,\ldots,w_i$，价值为$v_1,v_2,\ldots,v_i$。
  - 递推关系
    - 前i-1个物品能够放入承重为j的背包中的最大价值是多少？→ V[i-1,j] 和 V[i,j]之间的关系
    - 两个序列中分别的前i个和前j个字符中包括的最大公共子串和少一个字符的最大公共子串之间的关系

  - V[i,j]分为两种情况
    - 第i个物品不能放入背包：$j<w_i$ → V[i-1,j]
    - 第i个物品能放入背包：$j >= w_i$
      - 不放第i个物品的最大价值是V[i-1,j]
      - 放第i个物品的最大价值是v_i + V[i-1, j-w_i]

问题五：背包问题

- 边界：V[0,j] = 0; V[i,0] = 0

	0	$j\text{-}w_i$	j	W
0	0	0	0	0
i-1		$V[i-1,j-w_i]$	V[i-1,j]	
i			V[i,j]	
n				目标

- $$V[i,j]=\begin{cases} \max\{V[i-1,j],\ V[i-1,j-w_i] + v_i\}, & j>=w_i \\ V[i-1,j], & j<w_i \end{cases}$$

问题六：马尔可夫模型——定义

- 马尔可夫系统
  - 对有限自动机的扩展：状态集合、状态间转换关系集合（可加权）
  - 有一个有限的状态集合：$S=\{s_1,s_2,\ldots,s_{|s|}\}$
  - 有一个有限的时间序列：$t=\{1, 2, 3, \ldots, T\}$
  - 在每一个时间点，系统处于一个确定的状态$z_t \in \{s_1,s_2,\ldots,s_{|s|}\}$
  - 每个时间点的状态都是随机选择的
  - 当前时刻的状态决定了下一个时间点状态的概率分布
    - 状态转换概率矩阵：$A=\{a_{ij} \mid s_i \rightarrow s_j$转换的概率，$0<i$、$j<|S|\}$
    - 开始状态概率：$\pi \in R^{|S|}$

- 例子：天气变化模型，晴天、阴天、雨天、多云

问题六：马尔可夫模型——序列概率

- 给定一个马尔可夫系统，观测到$\vec{z} = \{z_1, z_2, \ldots, z_t, \ldots z_T\}$的概率
  - P($\vec{z}$)
  - =P($z_1, z_2, \ldots, z_t, \ldots z_T$)
  - =P($z_1, z_2, \ldots, z_t, \ldots z_{T-1}$)*P($z_T | z_1, z_2, \ldots, z_t, \ldots z_{T-1}$)
  - =P($z_1, z_2, \ldots z_{T-2}$)*P($z_{T-1} | z_1, z_2, \ldots, z_t, \ldots z_{T-2}$)*P($z_T | z_1, z_2, \ldots, z_t, \ldots z_{T-1}$)
  - ……
  - =P(z_1)*P($z_2|z_1$)*…*P($z_T | z_1, z_2, \ldots, z_t, \ldots z_{T-1}$)
  - = P(z_0)*P($z_1|z_0$)*P($z_1|z_0$)*…*P($z_t|z_{t-1}$)*…*P($z_T|z_{T-1}$)
  - =1*A_{z0z1}*$A_{z1z2} * \cdots * A_{zT-1zT}$
  - =$\prod_{t=1}^{T} A_{z_{t-1}z_t}$

问题六：马尔可夫模型——特定状态概率

- 给定模型，则t时刻观察到状态z_t=s的概率
  - P(z_t=s)
  - 设长度为t的状态序列Z是一个以z_t=s结尾的序列，则
  - P(z_t=s)=$\sum_{z\text{所有可能的情况}} \mathrm{P}(Z)$

- 蛮力解法
  - 罗列Z所有可能的序列
  - 求每一个P(Z)
  - 求和

- 巧妙解法：动态规划

问题六：马尔可夫模型——动态规划解法（1）

- 给定模型，则t时刻观察到状态z_t=s的概率
- 如果知道t-1时刻处于某个状态s_i的概率，那么z_t=s_j的概率如何表示？

t-2	**S1**	**S2**	**…**	**Sn**	**…**	**S|s|**
t-1	S1	S2	…	Sn	…	S|s|
t	S1	S2	…	Sj	…	S|s|

$$P\left(z_t = s_j\right) = \sum_{i=1}^{|s|} P\left(z_t = s_j \land z_{t-1} = s_i\right)$$

$$= \sum_{i=1}^{|s|} P\left(z_t = s_j | z_{t-1} = s_i\right) * P(z_{t-1} = s_i)$$

$$= \sum_{i=1}^{|s|} a_{ij} * P(z_{t-1} = s_i)$$

问题六：马尔可夫模型——动态规划解法（2）

- 表格扩展

t	P($z_t = s_1$)	…	P($z_t = s_i$)	…	P($z_t = s_{\|s\|}$)
0					
1					
…					
T					

$$P(z_t = s_j) = \sum_{i=1}^{|s|} a_{ij} * P(z_{t-1} = s_i)$$

$$P(z_1 = s_j) = \sum_{i=1}^{|s|} a_{01} * P(z_0 = s_i) \qquad P(z_0 = s_i) = \begin{cases} 1, & \text{以}si\text{为开始状态} \\ 0, & \text{不以}si\text{为开始状态} \end{cases}$$

作业题：两个字符串最小编辑距离

- 给定两个字符串A和B，求字符串A至少经过多少步字符操作变成字符串B。允许的操作有：
  - 删除一个字符
  - 插入一个字符
  - 修改一个字符

- 用edit[i][j]表示A串和B串的编辑距离

$$\mathrm{edit}[i][j]=\begin{cases}0 & i=0, j=0\\ j & i=0, j>0\\ i & i>0, j=0\\ \min(edit[i-1][j]+1, edit[i][j-1]+1, edit[i-1][j-1]+flag) & i>0, j>0\end{cases}$$

，其中

$$\mathrm{flag}=\begin{cases}0 & A[i]=B[j]\\ 1 & A[i]\neq B[j]\end{cases}$$

谢 谢！