A short note on monotone matrices and the sum selection problem

Asish Mukhopadhyay and Eugene Greene

June 10, 2008

Abstract

Given a sequence of n real numbers $a_1, a_2, a_3, \ldots, a_n$, the *maximum segment sum* problem is that of determining indices i and j $(1 \leq i \leq j \leq n)$ such that the sum $s(i,j) = a_i + a_{i+1} + \ldots + a_j$ is a maximum. Monotone matrices were shown to be remarkably effective in solving several geometric optimization problems. The surprise is that it can also be applied to the above problem as we show here.

1 Introduction

Given a sequence of n real numbers $a_1, a_2, a_3, \ldots, a_n$, the *maximum segment sum* problem is that of determining indices i and j $(1 \leq i \leq j \leq n)$ such that the sum $s(i,j) = a_i + a_{i+1} + \ldots + a_j$ is a maximum. This problem was introduced by Jon Bentley in his CACM column (#8) on Programming Pearls. He described an elegant linear time algorithm due to J. B. Kadane [Ben84].

Monotone matrices have been used to solve a variety of geometric optimization problems [AKM+87]. In this paper, we show how it can applied to solve the largest segment sum problem in linear time. In fact, this scheme allows us to find the largest segment sum for all start positions in the array, as well as the largest segment sum for a segment length that lies between specified length-parameters l and u, within the same time bounds.

2 Kadane's scheme

Kadane's scheme is based on the clever observation that a maximum segment sum cannot have a prefix with a negative sum, or, for that matter a suffix with a negative sum.

It finds the *start*, *end*, and $maxSum$ of a maximum segment sum, using three variables - two index variables i and j, with $i \leq j$ always and a $currentSum$ which is the segment sum from i to j. As j sweeps over the array, the values of *start*, *end* and $maxSum$ are updated whenever $currentSum$ exceeds $maxSum$. Whenever $currentSum$ becomes negative, the variable i jumps to $j+1$, since all segment sums with a start value in $[i..j]$ would have a negative prefix.

The time complexity of this scheme is clearly in $O(n)$.

3 Monotone matrix approach

An $n \times n$ matrix of reals is *monotone* if the maximum entry in row i occurs in the same column or in a column to the right of the column in which the maximum entry in row $i-1$ occurs. See Example 1 below.

A matrix is said to be *totally monotone* if for any 2×2 submatrix $\begin{bmatrix} a & b \\ c & d \end{bmatrix}$, it is not simultaneously possible that $a < b$ and $c > d$.

Let M be an $n \times n$ matrix such that $M[i][j] = s(i,j)$ when $i \leq j$, and $M[i][j] = -\infty$ when $i > j$.

Fact 1 *M is a totally monotone matrix.*

Proof 1 *Let indices i, j, k, and l be such that $1 \leq i < k \leq j < l \leq n$. Assume $M[i][j] < M[i][l]$. If $M[i][j] = -\infty$ then $M[k][j] = -\infty$ as well, and so $M[k][j] \leq M[k][l]$. Otherwise, $a_i + \ldots + a_k + \ldots + a_j < a_i + \ldots + a_k + \ldots + a_j + \ldots + a_l$. So $a_k + \ldots + a_j < a_k + \ldots + a_j + \ldots + a_l$, and hence $M[k][j] \leq M[k][l]$.*

By precomputing all prefix sums $P[i] = a_1 + a_2 + \ldots + a_i$ (we define $P[0] = 0$), we can compute an $M[i][j]$ in constant time as $M[i][j] = P[j] - P[i-1]$ for $i \leq j$.

Now we can use the monotone matrix searching results of [AKM+87] to determine the largest segment sum in each row in $O(n)$ time. Thus we have a largest segment sum beginning at a given index i ($1 \leq i \leq n$), while the largest of these is a maximum segment sum.

Example 1 *Consider the sequence 5, -10, 6, -10, 7, -10, 8. The monotone matrix, M, corresponding to this sequence is:*

	1	2	3	4	5	6	7
1	5	-5	1	-9	-2	-12	-4
2	-∞	-10	-4	-14	-7	-17	-9
3	-∞	-∞	6	-4	3	-7	1
4	-∞	-∞	-∞	-10	-3	-13	-5
5	-∞	-∞	-∞	-∞	7	-3	5
6	-∞	-∞	-∞	-∞	-∞	-10	-2
7	-∞	-∞	-∞	-∞	-∞	-∞	8

This approach allows us to solve a slightly more general version of the segment sum problem. We can find the maximum segment sum when the length of such a segment is restricted to lie between l and u, where $1 \leq l \leq u \leq n$, by only considering entries along the diagonals defined by $j - i + 1 = l, l+1, \ldots, u$ and setting all other entries to $-\infty$.

4 Conclusions

We have described an interesting connection between monotone matrices and the problem of finding a maximum segment sum of a sequence of n numbers. This gives us another $O(n)$ algorithm, which albeit more complicated is also more general and powerful.

We have implemented this algorithm and the interested reader can try it out by clicking on the software link at `www.cs.uwindsor.ca/~asishm`.

References

[AKM+87] Alok Aggarwal, Maria M. Klawe, Shlomo Moran, Peter W. Shor, and Robert E. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2:195–208, 1987.

[Ben84] Jon Bentley. Programming pearls: algorithm design techniques. *Commun. ACM*, 27(9):865–873, 1984.