EE447 Homework

付昊源 517021910753

March 29, 2020

Problem Statement

Suggest there are node A and B in this space, between which exists n linked edges. The existing probability of the i^{th} edge is p_i where $i = 1, 2, \cdots n$. And the cost of detecting whether the i^{th} edge exists is c_i. The graph is denoted as G_n, shown in Figure 1. In this problem, we need to design a strategy to make sure the total cost for detecting is optimal in expectation, and prove its optimization.

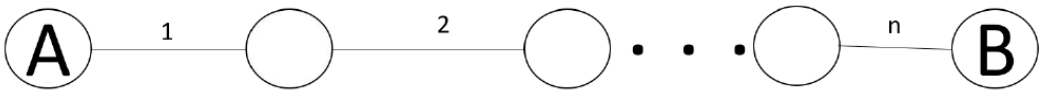

图 1: The linked-edge graph Gn including node A and B

Solution

Problem Analysis

Accoring to the problem statement, the total cost is related to the order that we detect each edge. Suggest we detect all n edges in order $< s_1, s_2, \cdots, s_n >$, where set $\{s_1, s_2, \cdots, s_n\}$ is equivelent to $\{1, 2, \cdots, n\}$. In this situation, the expectation of total cost $\mathbf{E}(C(< s_1, s_2, \cdots, s_n >))$ is

$$
\begin{aligned}
&\mathbf{E}(C(< s_1, s_2, \cdots, s_n >)) \\
=&(1-p_{s_1})c_{s_1} + p_{s_1}(1-p_{s_2})(c_{s_1}+c_{s_2}) + \cdots + (\prod_{i=s_1}^{s_{n-1}} p_i)(1-p_{s_n})(\sum_{j=s_1}^{s_n} c_j) + (\prod_{i=s_1}^{s_n} p_i)(\sum_{j=s_1}^{s_n} c_j) \\
=&c_{s_1} + c_{s_2}p_{s_1} + c_{s_3}p_{s_1}p_{s_2} + \cdots + c_{s_n}\prod_{i=s_1}^{s_{n-1}} p_i \\
=&\sum_{i=0}^{n-1}(c_{s_{i+1}}\prod_{j=0}^{i} p_{s_j}) \quad \text{where } p_{s_0} = 1
\end{aligned}
\tag{1}
$$

In this problem, we want to translate edges $\{1, 2, \cdots, n\}$ into an order $< s_1, s_2, \cdots, s_n >$ such that $\mathbf{E}(C(< s_1, s_2, \cdots, s_n >))$ is optimal. In next subsection, I will first give the algorithm which can make

the total cost of edge detection optimal in expectation. And in the last subsection, I will give a proof to the optimization of my algorithm.

Algorithm

Algorithm 1 Optimal order for edge detection

Input: $p = [p_1, p_2, \cdots, p_n]$ and $c = [c_1, c_2, \cdots, c_n]$

Output: An order of elements $1, 2, \cdots, n$

$s = [(\frac{c[i]}{1-p[i]}, i+1)$ for i in range$(0, n)]$

$res \leftarrow$ empty list

Sort list s according to elements $s[i][0]$ in increasing order.

For i in range$(0, n)$:

    res.append$(s[i][1])$

Return res

Proof

Consider any given order $< s_1, s_2, \cdots, s_n >$, we know that

$$\mathbf{E}(C(< s_1, s_2, \cdots, s_n >)) = \sum_{i=0}^{n-1}(c_{s_{i+1}} \prod_{j=0}^{i} p_{s_j}) \tag{2}$$

from equation 1. And let's first swap s_i and s_{i+1} (where $1 \leq i \leq n-1$), then compare the expectation of these two orders. For ease to write, let's denote the the order before swapping as o_1 and the order after swapping as o_2. We can easily get

$$\begin{aligned}
&\mathbf{E}(C(o_1)) - \mathbf{E}(C(o_2)) \\
=&(\prod_{j=0}^{i-1} p_{s_j})(c_{s_i} + p_{s_i} c_{s_{i+1}} - c_{s_{i+1}} - p_{s_{i+1}} c_{s_i}) \\
=&(\prod_{j=0}^{i-1} p_{s_j}) \left[(1 - p_{s_{i+1}})c_{s_i} - (1 - p_{s_i})c_{s_{i+1}}\right] \\
=&(\prod_{j=0}^{i-1} p_{s_j})(1 - p_{s_i})(1 - p_{s_{i+1}})(\frac{c_{s_i}}{1 - p_{s_i}} - \frac{c_{s_{i+1}}}{1 - p_{s_{i+1}}})
\end{aligned} \tag{3}$$

Thus, we know that if term $\frac{c_{s_i}}{1-p_{s_i}}$ is smaller, then $\mathbf{E}(C(o_1))$ is better and vice versa.

To compare any given order o and our optimal order o^*, we can consider o as swapping finite number of elements in o^*. And for each pair of swapping elements, it is equivalent to swap each adjacent elements between the pair. For each swap with s_i and s_{i+1}, according to my algorithm, we have

$$\frac{c_{s_i}}{1 - p_{s_i}} \leq \frac{c_{s_{i+1}}}{1 - p_{s_{i+1}}} \tag{4}$$

since in o^*, index are sorted by term $\frac{c_{s_i}}{1-p_{s_i}}$. And according to previous proof, after swapping, the expectation of total cost will be equal or higher, depending on whether the equals sign in equation 4 satisfies.

In conclusion, for any given order to detect the edges, the expectation of total cost is higher than that of optimal order. And the factor in deciding which edge should be detected earlier is $\frac{c_i}{1-p_i}$, if this term is smaller, that edge should be detected earlier.