Information Processing in Japan Vol. 17, 1977

Probable Transition Searching System

Yukuo ISOMOTO* and Keiichi YAMAGATA**

Abstract

A graph theory and a probability theory are very useful to discuss transition phenomena. We are interested in applying these theories to an information retrieval system which is represented by a weighted directed graph. This paper shows its theoretical foundation and its general property. Using the formula, we have realized the probable transition searching system with a FORTRAN program. The system can efficiently retrieve the most probable sequence of events which are connected with one another by a binary relation. If you can ideally abstract the binary relation and its weight from a concrete object, the model is applicable to any objects in virtue of the general technique.

1. Introduction

There are many logocal systems whose performance depends not only on the characteristics of events but also on the relation of events[1,2]. The relation is represented by means of a graph. We are concerned here with the property of the system which retrieves the most probable sequence of events from many possible ones and discuss how to formulate and realize the system.

The model is represented by a weighted directed graph on the assumption of a weighted binary relation, which is so called a transition probability in Markov process. A typical application of this system is a search for a complex causal relation of events. The causality often depends on the circumstances. However, the causal relation could be discussed with statistical probabilities if neglecting the dependence. And then these data can be treated by this system well.

2. Notation and Formula

On the analogy of causal relations, this chapter devotes to formulate the system retrieving the most probable sequence of events in a pseudo order set. Let V be a

---

This paper first appeared in Japanese in Joho-Shori (Journal of the Information Processing Society of Japan), Vol. 17, No. 8 (1976), pp. 695~702.

* Osaka University Computation Center, Osaka University

** Faculty of Engineering, Osaka University

finite set of events $\{v_1, v_2, \ldots, v_m\}$ which are called vertices in a graph. Corresponding to a causality, an 1-step transition from v_i to v_j is denoted by a binary relation $v_i R v_j$ and called an arc in a graph. v_i or v_j are called a successor or a predecessor of each other. The set of all successors of v_i is denoted by $\Gamma(v_i)$ and the set of all predecessors of v_i by $\Gamma^{-1}(v_i)$. For $U \subset V$, let $\Gamma(U) = \bigcup_{v_i \in U} \Gamma(v_i)$, $\Gamma^{-1}(U) = \bigcup_{v_i \in U} \Gamma^{-1}(v_i)$. Similar to $\Gamma(v_i)$, a subset of reachable vertices from v_i after a n-step transition $\Gamma^n(v_i)$ is defined as $\Gamma^0(v_i) = \{v_i\}$, $\Gamma^1(v_i) = \Gamma(v_i)$ and $\Gamma^n(v_i) = \Gamma(\Gamma^{n-1}(v_i))$, $n \geq 2$. Moreover, a subset of all reachable vertices from v_i is defined as $\hat{\Gamma}(v_i) = \bigcup_{n=0}^{\infty} \Gamma^n(v_i)$. The pseudo order relation is defined as $v_j \lesseqqgtr v_i$ for $v_j \in \hat{\Gamma}(v_i)$. $\hat{\Gamma}(v_i)$ shows us the whole of events which are connected to v_i by any causal chains.

For multiple branch transitions, transition probabilities p_{ij} satisfy the condition:

$$\sum_{v_j \in \Gamma(v_i)} p_{ij} = 1, \qquad 0 \leq p_{ij} \leq 1. \tag{2.1}$$

p_{ij} depends only on the binary relation, and then the graph $G(V)$ is a Markov chain graph. For a transition matrix $P = [p_{ij}]$, a distribution vector $\vec{q}(n)$ after n-step transition is calculated by $\vec{q}(n) = \vec{q}(0) P^n$. The initial probability is assumed to be $q_s(0) = 1$ and $q_i(0) = 0, (i = 1, 2, \ldots, s-1, s+1, \ldots, m)$: v_s is a source of causal chains.

We call C_k a cluster if the subgraph $G(C_k)$ is strongly connected, where k means the equivalent class number of $\lesseqqgtr$. As our central problem concerns a pseudo order set, for simplicity, we reduce a closed cluster satisfying $\Gamma(C_k) = C_k$ to an end vertex with a self-loop. But the reduction of closed clusters does not affect distribution probabilities of all other vertices.

After the reduction of closed clusters, it is enough to consider $V' \subset \hat{\Gamma}(v_s)$ which is composed of a transient subset V'_T and an absorbing subset V'_E: $V' = V'_T + V'_E$. Rearranging elements of a transition matrix and a distribution vector in such order as V'_T and V'_E, the following expressions are obtained,

$$P' = \begin{array}{c} \left[\begin{array}{cc} P'_T & P'_E \\ 0 & I \end{array}\right] \begin{array}{l} \}r \\ \}m'-r \end{array} \end{array}, \qquad \vec{q}(n) = [\ \vec{q}'_T(n),\ \vec{q}'_E(n)\ ], \tag{2.2}$$

where I is a $(m'-r) \times (m'-r)$ unit matrix.

For all paths from a starting vertex v_s to an end vertex v_e, the total reachable probability b_{se} is given by the (s,e) element of the matrix B without trying to trace all possible paths[3] :

$$B = (I - P'_T)^{-1} P'_E . \tag{2.3}$$

If $v_e \in V'_E$ is an end vertex of one of causal chains starting from v_s, b_{se} gives the strength of causal relation between v_s and v_e.

Now, what is the most expectant path from v_s to v_e? In order to determine the most probable path from v_s to v_e, the product of transition probabilities is considered along a path. The largest product is named the maximum transition probability $\Pi_s(v_e)$ and its path is named the maximum transition probability path $F_s(v_e)$. $F_s(v_e)$ means the most probable causal chain of the paths from v_s to v_e. $\hat{\Gamma}(v_s)$, b_{se} and $F_s(v_e)$ give us the information about the most expectant causality among events in V.

3. Computational Technique

This chapter shows efficient computational techniques for the followings: (1) $\hat{\Gamma}(v_s)$, (2) Cluster C_k in $\hat{\Gamma}(v_s)$, (3) b_{se} for $v_e \in V_E'$ and (4) $\Pi_s(v_e)$ and $F_s(v_e)$. These are the essential factors in the complex causal relation.

3.1) Reachable Set $\hat{\Gamma}(v_s)$ for a Starting Vertex v_s

In order to avoid double counts of a vertex, we define the recursive formula :

$$\Lambda_0(v_s) = \{v_s\}, \quad \lambda_1(v_s) = \{v_s\}^c \cap \Gamma(v_s), \quad \Lambda_1(v_s) = \{v_s\} \cup \Gamma(v_s). \qquad (3.1)$$

$$\lambda_n(v_s) = \Lambda_{n-1}^c(v_s) \cap \Gamma(\lambda_{n-1}(v_s)), \quad \Lambda_n(v_s) = \Lambda_{n-1}(v_s) \cup \lambda_n(v_s), \ n \geq 2, \qquad (3.2)$$

where Λ_{n-1}^c and $\{v_s\}^c$ are complements of Λ_{n-1} and $\{v_s\}$ respectively. $\lambda_n(v_s)$ is composed of only vertices which are not reachable until n-step transition. $\Lambda_n(v_s)$ is composed of all elements in the range of n-step transitions and satisfies the equation :

$$\Lambda_n(v_s) = \bigcup_{\nu=0}^{n} \Gamma^{\nu}(v_s), \quad n \geq 2. \qquad (3.3)$$

The sequence $\Lambda_1 \subset \Lambda_2 \subset \ldots \subset \Lambda_n$ has an upper limit at the N-step. Thus we obtain

$$\hat{\Gamma}(v_s) = \lim_{n \to \infty} \Lambda_n(v_s) = \Lambda_N(v_s), \quad \lambda_N(v_s) = \emptyset. \qquad (3.4)$$

If Γ^{-1} is used for Γ in eqs.(3.1) and (3.2), we obtain a set $\check{\Gamma}(v_s)$: a transition starting from $v_j \in \check{\Gamma}(v_s)$ reaches v_s without exception.

3.2) Partition of Clusters and Total Reachable Probability b_{se}

Clusters are partitioned through two steps. A vertex in a cluster must have both of two arcs incident into and out of itself at least. Then the v_i is removed from $\hat{\Gamma}(v_s)$ with its arcs if v_i has no arc incident into or out of it besides a self-loop. By repeating such a procedure, we can obtain a subset S which is composed of some clusters and vertices having arcs incident into a cluster and out of another.

v_i and v_j in the same cluster must satisfy the relations $v_i \leq v_j$ and $v_j \leq v_i$, which are equivalent to the conditions $v_j \in \check{\Gamma}(v_i)$ and $v_j \in \hat{\Gamma}(v_i)$. Thus a cluster C is obtained as

$$C = \check{\Gamma}(v_i) \cap \hat{\Gamma}(v_i) \ . \qquad (3.5)$$

However C is not a cluster if C has less than two vertices. Other clusters are partitioned from the subset S'=S-C in the same way as eq.(3.5). Finally, closed clusters are replaced by end vertices. Then P' in eq.(2.2) is obtained after the replacement of closed clusters and b_{se} is easily calculated with eq.(2.3).

3.3) Maximum Transition Probability $\Pi_s(v_e)$ and its Path $F_s(v_e)$

Consider a product of probabilities $\Pi_i(v_e)=p_{ij}p_{jk}\cdots p_{le}$ along one of the paths from $v_i \in V'_T$ to v_e. $\Pi_s(v_e)$ is obtained as all $\Pi_i(v_e)$ satisfy the equation,

$$\Pi_i(v_e) = \max_{v_j \in \Gamma(v_i)} \left[p_{ij} \cdot \Pi_j(v_e) \right], \qquad v_i \in V'_T, \qquad \Pi_e(v_e) = 1. \qquad (3.6)$$

Eq.(3.6) is solved by the successive approximation[4] as

$$\Pi_i^{(1)}(v_e) = p_{ie} \text{ for } v_i \in \Gamma^{-1}(v_e) \quad \text{and} \quad \Pi_i^{(1)}(v_e) = 0 \text{ for } v_i \notin \Gamma^{-1}(v_e). \quad (3.7)$$

$$\Pi_i^{(k)}(v_e) = \max_{v_j \in \Gamma(v_i)} [p_{ij} \cdot \Pi_j^{(k-1)}(v_e)]. \qquad (3.8)$$

$\Pi_i^{(k)}(v_e)$ means k-th approximation of $\Pi_i(v_e)$. When $\Pi_i^{(K)}(v_e)=\Pi_i^{(K-1)}(v_e)$ for all v_i, $\Pi_s(v_e)$ is obtained: $\Pi_s(v_e) = \Pi_s^{(K)}(v_e)$. (3.9)

The sequence of vertices satisfying eq.(3.9) is $F_s(v_e)$ corresponding to the $\Pi_s(v_e)$.

4. Binary Data Manipulation for Sets in a Computer

Original data are stored in the data file as shown in Fig.4.1. v_i, $\Gamma(v_i)$ and p_{ij} are array elements with a subscript i. They are transfered to the main memory in order of transitions. The sets $\lambda_n(v_s)$, $\Lambda_n(v_s)$ and $\Gamma(v_i)$ are manipulated by masking technique.

```
                :
(IIII)J/MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM...
/KKK1/XXXXXXXXXX//KKK2/YYYYYYYYYY//...
                :
   IIII; I4, a code number for an event.
      J; I1, the number of transitions from IIII.
 MM...M; 8A8, a message for the event IIII.
   KKK1; I4, a code number for a successor of IIII.
 XX...X; F10.5, a transition probability from IIII
         to KKK1.
   KKK2; I4, a code number for a successor of IIII.
 YY...Y; F10.5, a transition probability from IIII
         to KKK2.
```

Fig. 4.1 Data format in the data file

In Fig.4.2, v_1 and $\Gamma(v_1)$ are transfered to the main memory at first. The sets $\Gamma(v_1)$, $\Lambda_1(v_1)$ and $\lambda_1(v_1)$ are expressed with bit strings:

$\Gamma(v_1)$: 011000... , $\Lambda_1(v_1)$: 111000... , $\lambda_1(v_1)$: 011000... .

The value 1 of a bit in the arrays means the existance of the corresponding vertex. For example, $\Gamma(v_1)$ denotes the transition from v_1 to v_2 and v_3. $\Lambda_1(v_1)$ is composed of v_1, v_2 and v_3 in the range of 1-step transition. The 2nd and 3rd bits in $\lambda_1(v_1)$ denote that v_2 and v_3 are new elements in $\Lambda_1(v_1)$. At the second step, the computer searches v_2 and v_3 in the data file according to $\lambda_1(v_1)$. $\Gamma(v_2)$ and $\Gamma(v_3)$ are transfered to the main memory. Then the following values are given as, $\lambda_2(v_1)$: 00011100... ,

$\Lambda_2(v_1)$: 11111100... , $\Gamma(v_2)$: 00111000... , $\Gamma(v_3)$: 00001100... .

The 1 in the 4th, 5th and 6th bits in $\lambda_2(v_1)$ means that v_4, v_5 and v_6 are new elements in $\Lambda_2(v_1)$. These procedures are iteratively executed until $\hat{\Gamma}(v_1)$ is finally obtained as $\lambda_N(v_1)=\emptyset$. The transition probabilities and the messages about vertices are transfered to the main memory at the same time as v_i and $\Gamma(v_i)$. All other sums and products of sets in chapter 3 are also calculated by the masking technique.

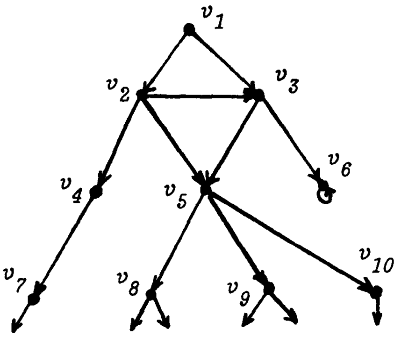

Fig. 4.2 A directed graph

5. Conclusion

We have many phenomena which can be often simulated by probable transition processes among events in a pseudo order set. This paper shows some formulae for the probable transition searching system. FORTRAN program debugging process is a good example for this system. A starting event v_s corresponds to a detected phenomenon about a bug in a program. $v_i \in V'_T$ are associated phenomena of the bug or interpretations for the aid of a programmer's understanding about the bug. $v_e \in V'_E$ is a final debugging procedure or debugging technique. The graph associated with v_i and $\Gamma(v_i)$ gives us some possible processes of the debugging. Owing to the abstracted formula, its applications are free from the concrete meaning of the data.

This probable transition searching system can analyze a weighted directed graph with 50 vertices in a few seconds. The short analyzing time gives us an assurance of the practical use of this system. This system has been practically applied to the Information Service System for Program Debugging.[5]

References

1) Salton G.: Automatic Information Organization and Retrieval (Computer Science Series), McGraw-Hill Book Company, (1968).

2) Ernst G. W. and Newell A.: ACM Monograph Series, GPS: A Case Study in Generality and Problem Solving, Academic Press, (1969).

3) Kemeny J. G. and Snell J. L.: Finite Markov Chains, D. Van Nostrand Company, INC. p.52, (1960).

4) Bellman R., Cook K. L. and Lockett J.: Algorithm, Graph and Computers, Mathematics in Science and Engineering, Academic Press, Vol.62, pp.49-100, (1970).

5) Makinouchi S., Isomoto Y. and Yamagata K.: Information Service System for Program Debugging, Technology Reports of the Osaka University, Vol.27, pp.211-219, (1977).