Alignment-based Similarity Measurement for Action Traces

Juntao GAO, Danchun ZHAO, Hongbo ZHOU, Yongan WANG

School of Computer and Information Technology, Northeast Petroleum University, Daqing, 163318 China

Summary
Similarity measurement of action traces is a basic operator which is useful in several scenarios during business process management, such as process mining, process model search, process reengineering and so on. Firstly, according to an information-theoretic definition of similarity, the reference measure of action traces similarity is defined; secondly, the technologies of semantic computing are employed to address ambiguity issues caused by the use of synonyms or homonyms; thirdly, Hungarian algorithm is extended to reduce the time complexity of picking out the best match from similarity matrix; fourthly, approximate longest common sub-trace is defined to identify the commonality of traces. Finally, the trace similarity is constructed and an experiment is given to evaluate the method.

Key words:
semantic similarity, action traces, firing sequence, business processes.

1. Introduction

Along with the wide application of process-aware information systems, such as ERP (Enterprise Resource Planning), SCM (Supply Chain Management), PDM (Product Data Management), large amount of action traces have been accumulated in various information systems. An action trace, also called firing sequence in the domain of Petri nets, is a finite or infinite sequence of activities that denotes the order in which the execution of activities starts in an instance of the process.

These action traces are important intellectual assets of organizations, so a deep insight into these action traces and their mutual relationship is necessary to business process management activities. There are various applications in business process management that require measuring the similarity between action traces, such as process mining, process model search, process reengineering and so on. For example, the sets of action traces of business processes are compared to calculate compliance and maturity of an actual process model to a reference model in process reengineering [1]. In this context, the compliance degree and the maturity degree of two traces are defined based on their longest common subsequence. After that, the overall compliance and maturity degree between two models are calculated by summing up the maximum compliance and maturity degree of traces belong to them. Another example is to align action traces in quantitative analysis method of business process [2]. The actual action traces are compared to the simulated traces from predefined business process models. According to the result of action trace comparing, the bottleneck and critical path are identified. In paper [3], the set of traces are used to construct a reference similarity of business processes. In this paper, a method based on alignment technology is proposed to measure semantic similarity of action traces which is essential to action traces analysis.

This paper is constructed as follows: In the next section, the state-of-the-art methods to measure action traces are reviewed. The similarity measurement of action traces is constructed in section 3. Section 4 discussed the ambiguity issues in alignment of action traces and the way to construct similarity matrix and discussed the time complexity to identify the best match from similarity matrix and import Hungarian algorithm to reduce the time complexity. Section 5 presents the approximate longest common sub-traces to identify the commonality of traces and apply the method to the exemplary action traces to evaluate the method. At last, the conclusion is drawn.

2 The State of The Art

Although several approaches have recently been proposed to measure the similarity between action traces, neither the definitions of the similarity notion between traces nor the measure methods have gained wide recognition. Four kinds of methods based on editing distance were proposed to compute the difference between action traces [2]. The first method employs Hamming distance to compare action traces, which does not applied to traces of unequal size. The second one is based on simple editing distance, which allows traces unequal of size. But it does not take into account the differences among the weight of editing operations. The third one and the fourth one adopt weight to define the cost and importance of editing operations without presentation that how to determine the value of weight. Therefore, the two methods have considerably less practicality. The longest common subsequence was proposed to measure the similarity of traces [1]. The problem of longest common subsequence is NP-hard for the general case of an arbitrary number of input action traces [4]. Even though the complexity can be polynomial time by dynamic programming algorithm when the size of

Manuscript received August 5, 2014
Manuscript revised August 20, 2014

action traces is constant, the method is too rigid for measurement of action trace similarity.

The above researches focus on the action traces in which each action is a symbol or numeric value. However, the real action traces are often described by a short text. The relationship between these actions is often not identical or completely different. Further more, the same action traces may be represented in different ways in deferent organizations. This is because that different organization using different terminologies and the problem of semantic heterogeneity makes it a tedious job to compare textual action traces. In this paper, a reference definition of action trace similarity is constructed according to information theory. The idea of similarity propagation is introduced to pick out a mapping between corresponding activities and data, and Hungarian algorithm is expanded to reduce its time complexity. Then the similarity of whole models is measured based on Jaccard coefficient. Finally, an experiment is given to evaluate the method.

3 Definition of Similarity

Before the formal definition of the intuitive concept of similarity is provided, the intuitions about similarity are first clarified.

3.1 The Intuitions

Intuition 1:
The similarity between two traces is related to their common actions. The more common actions they share, the more similar they are.

Intuition 2:
The similarity between two traces is related to their different actions. The more common different actions they have, the less similar they are.

Intuition 3:
The maximum similarity between two traces is reached when they are identical, no matter how many common actions they share.

Intuition 4:
If there are no common actions but similar actions from two traces, then the two action traces are not completely different.

Intuition 5:
If the action sets from two traces are identical, but the orders are completely different, then the two action traces are not completely different.

Intuition 5:
The commonality of action sets has a larger effect than the same order between action traces.

From the above assumptions, we can prove the following theorem.

3.2 Reference Similarity

In this section, a reference similarity is discussed to satisfy the above intuitions. Suppose Σ denotes the universe of actions, there exist two traces α and β,

$$\alpha =< x_1, x_2, ..x_m > \text{ and } x_i \in \Sigma, i \in [1, m]$$
$$\beta =< y_1, y_2, ..., y_n > \text{ and } y_j \in \Sigma, j \in [1, n]$$

x_i denotes the i-th action in trace α, i=1,2,...,n. $|\alpha|$ denotes the length of trace α, which is the number of actions in α. y_j denotes the j-th action in trace β, j=1,2,...,n. $|\beta|$ denotes the length of trace β, which is the number of actions in β.

The commonality of α and β is depicted by $common(\alpha, \beta)$,

$common(\alpha, \beta) =< X \cap Y, lct >$, X is the action set of α, Y is the action set of β. Because the action may occur more than once, X and Y are both multisets. The commonality of α and β includes two parts: one is the common action set $X \cap Y$, the other is the longest common subtrace, lcs for short, from the common action set. It is deployed to measure the similarity of the order of action occurring.

The combination of trace α and β is depicted by $description(\alpha, \beta)$,

$description(\alpha, \beta) =< X \cup Y, \{\alpha, \beta\} >$,

The combination of α and β also includes two parts, one is the union of action set $X \cup Y$, the other is two alternative action sequences $\{\alpha, \beta\}$.

According to information theory [5], the reference similarity of action traces is:

$$sim(\alpha, \beta) = \frac{\log P(common(\alpha, \beta))}{\log P(descritipn(\alpha, \beta))} \quad (1)$$

If the probability of trace is known, the above formula can be computed using the following formula.

$$sim(\alpha, \beta) = \sqrt{\varepsilon \times \left(\frac{|X \cap Y|}{|X \cup Y|}\right)^2 + \varphi \times \left(\frac{|lct|}{|X \cap Y|}\right)^2} \quad (2)$$

where, $\varepsilon \geq 0, \varphi \geq 0$ and $\varepsilon + \varphi = 1$.

The value of ε and φ is determined by the amount of information contained in the action sets and their orders. Generally, the cardinality of universal action set is very large, so the probability of common actions occurring is very little and the amount of information contained in action sets is very large. While given the common action

set, the probability of the same order occur is relatively big and so the amount of information contained in it is relatively less. Therefore, ε is bigger than φ.

4 Computing the Commonality of Action Set

Similar traces are defined based on the action similar. In reality, the action trace is identified by short text. The result of comparison between action x_i and y_j is not a binary value. Therefore, the traditional method to compute the union and intersection of action sets does not work in this case. In this section, an alignment-based method is discussed to compute the commonality of action set. This method involves three steps: (1) construct the similarity matrix; (2) pick up best matching; (3) computing the commonality. Next, each step is going to be explained.

4.1 Constructing the Similarity Matrix

Because the identifiers of action may be made by different systems, the vocabulary employed to identify actions may well be different. Therefore, the synonyms and homonyms are inevitable and make it difficult to compare the actions. In order to address the semantic heterogeneity, edit distance [6] and WordNet [7] is combined to measure the initial action similarities, which is the seeds of similarity matrix to start the iteration. Next, the process iteration is presented.

Given two actions $x \in \alpha$ and $y \in \beta$, the semantic similarity is defined as $SimA(x, y) \in [0,1]$. It is a total function over $\alpha \times \beta$ and determined by an iterative computation to simulate the flooding phenomenon of similarity among action traces. The flooding phenomenon means that if x_i is much similar to y_j then the similarity between x_{i-1} and y_{j-1} raise, as well as the similarity between x_{i+1} and y_{j+1}

$$SimA_k(x_i, y_j) = \omega SimA_{k-1}(x_i, y_j) + \varphi SimA_{k-1}(x_{i-1}, y_{j-1}) + \lambda SimA_{k-1}(x_{i+1}, y_{j+1}) \quad (3)$$

If $i = 0$ or $j = 0$, there is no pre-action of action x_i or no pre-action of action y_j. Therefore,

$$SimA_k(x_i, y_j) = \omega SimA_{k-1}(x_i, y_j) + \lambda SimA_{k-1}(x_{i+1}, y_{j+1}) \quad (4)$$

If $i = m$ or $j = n$, there is no sub-action of action x_i or no sub-action of action y_j. Therefore,

$$SimA_k(x_i, y_j) = \omega SimA_{k-1}(x_i, y_j) + \varphi SimA_{k-1}(x_{i-1}, y_{j-1}) \quad (5)$$

The similarity of action pair (x_i, y_j) is determined by the last result of iterative computation, its pre-action and sub-action. Fox example, if the names of two actions are different, but their pre-action and sub-action are same, their behavior must be more similar than their names. On the contrary, if their name seems alike, but pre-action and sub-action are absolutely different, their behavior must be less similar than their names. The first action in a trace has no pre-action and the last action in a trace has no sub-action. Therefore, the algorithm to compute these two kinds of action similarity is different from the ordinary actions. α, φ, λ are called propagation coefficients ranging from 0 to 1. They can be computed in many different ways.

After once flooding computation, the sum of similarity may shift a little. In order to keep the invariance of the sum of similarity, the result should be normalized, using the following formula.

$$SimA_k(x_i, y_j) = SimA_k(x_i, y_j) \times \frac{\sum_{i\in[1,m],\, j\in[1,n]} SimA_k(x_i, y_j)}{\sum_{i\in[1,m],\, j\in[1,n]} SimA_{k-1}(x_i, y_j)} \quad (6)$$

The computation is performed iteratively until the Euclidean length of the residual vector $\Delta(Simx_n, Simy_{n-1})$ becomes less than ε for some $n > 0$. If the computation does not converge, it is terminated after a certain number of iterations. The final similarity of actions is denoted as $SimA(x_i, y_j)$.

4.2 Picking up the best Matching

In the last section, all action pairs are assigned values to denote similarities. This section focuses on the issue how to pick out the best matching M, which maximizes the sum of similarity degrees. A mapping is a subset of activity pairs (x_i, y_j), in which x_i is from trace X and y_j is from trace Y. The combinatorial explosion of the number of mappings makes the issue difficult to resolve. Therefore, Hungarian algorithm [8] is expanded to solve the problem Here, the validity of the algorithm is discussed.

(1) Constructing similarity matrix. Computing the similarity of the action x_i in trace α and the action y_j in trace β .Then assigning the value to the element (i, j) in similarity matrix.

(2) Subtracting off the row min from each row.

(3) Subtracting off the column min from each column.

(4) Starting with the row or column with the least number of zeros, marks one certain zero element and redlines the row and the column where the marked zero element exists.

(5) Repeating step 4 until each zero element is marked or redlined. If the number of marked zero elements is $min(m, n)$, match the action of trace α in the row in which the zero element exists to the action of trace β in the column in which the zero element exists. Otherwise, go to step 6.
(6) Mark all rows without marked zero with *, and then mark all zero elements in rows with *, and mark all zero elements in columns with mark *, until mark * can not be added.
(7) Redline all the rows and columns without *.
(8) Identify the least one among the elements uncovered by lines and denoted by x_{ij}.
(9) Subtract x_{ij} from the rows marked with *, and subtract x_{ij} from the rows marked with *, return to step 4.

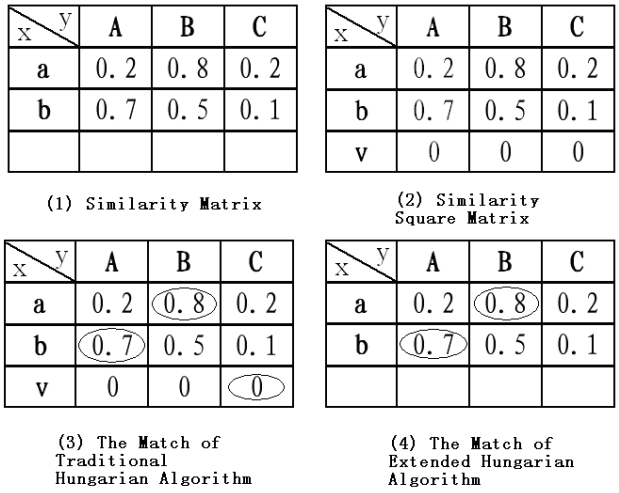


Fig. 1 the extended Hungarian algorithm

Theorem: given a bipartite graph $G = (X, Y, E)$ and its perfect matching M with maximum sum of weight. E_0 denotes the set of edges whose weight is zero. X_0 denotes the set of vertexes which are covered by E_0. Then $M' = M - E_0$ is the perfect matching of bipartite graph $G' = (X', Y, E')$. $X' = X - X_0$ and $E' = E - E_0$.

Proof: Next the proof of the theorem is given. If M' is not the maximum sum of weight, $\exists M'' \mid Sum(M'') > Sum(M')$, $Sum(M)$ is the sum of weight of M. M'' added with $|E'|$ zero-weight edges to cover the vertexes in X' is M'''. Because the difference between M'' and M''' are zero-weight edges, $Sum(M'') = Sum(M''')$. Similarly, $Sum(M) = Sum(M')$, so $Sum(M) < Sum(M''')$. It conflicts with the assumption.

4.3 Computing the Commonality

Given two action traces α and β:

$\alpha =< x_1, x_2, ..x_m >$ and $x_i \in \sum, i \in [1, m]$

$\beta =< y_1, y_2, ..., y_n >$ and $y_j \in \sum, j \in [1, n]$

The commonality is,

$$Common(\alpha, \beta) = \sum_{(x_i, y_j) \in M} SimA(x_i, y_j) \quad (7)$$

Obviously, $Common(\alpha, \beta)$ increases with the number of matched actions increasing and decreases with the number of matched activities decreasing. The value of $Common(\alpha, \beta)$ is not less than 0, which does not represent the similarity of action traces α and β, because it does not take the order of actions into account. Next, the algorithm to compare the order of action traces is discussed.

5. Measuring the Similarity of Action Traces

Using dynamic programming technology, such as Needleman-Wunsch Algorithm [9] and Smith–Waterman algorithm [10] the longest common subtrace is determined. The length of longest common sub-traces between action traces α and β is denoted as $lct(\alpha, \beta)$. If there is no common sub-traces between α and β, $lct(\alpha, \beta) = 0$. If α is same as β, $lct(\alpha, \beta) = len(\alpha) = len(\beta)$,where $len(\alpha)$ denotes the length of action trace α.

According to the best matching M, the scores for aligned actions are computed as following formula.

$$lct(x_i, y_j) = \begin{cases} lct(x_{i-1}, y_{j-1}) + 1 & (x_i, y_j) \in M \\ Max(lct(x_{i-1}, y_{j-1}), & \\ lct(x_{i-1}, y_j), lct(x_i, y_{j-1})) & (x_i, y_j) \notin M \end{cases} \quad (8)$$

Then the length of longest common sub-traces between action traces α and β can be computed using classical Needleman-Wunsch Algorithm.

Adopting the result of measure commonality depicted above, the similarity of action traces drills down to the following algorithm.

$\alpha =< x_1, x_2, ..x_m >$ and $x_i \in \sum, i \in [1, m]$

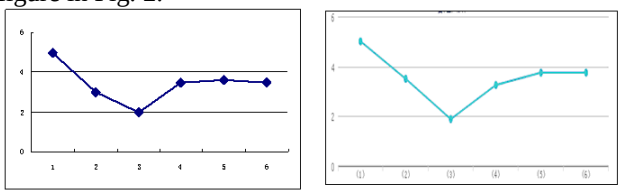
$\beta =< y_1, y_2, ..., y_n >$ and $y_j \in \sum, j \in [1, n]$

$$sim(\alpha, \beta) = \sqrt{\varepsilon \times SimASet(\alpha, \beta)^2 + \phi \times \frac{|lct|}{|\alpha \cap \beta|}^2} \quad (9)$$

Although there is no standard way to evaluate computational measures of model similarity, one reasonable way to judge can be agreement with human similarity ratings.

In the experiments, twenty subjects was chosen and given 6 trace pairs. The subjects were all experienced clerks in oil field. This target action trace is "a-b-c-c-d-e", the others 6 traces are （1）a-b-c-c-d-e,（2）a-b-c-d-e,（3）m-n-j-k-l-n,（4）e-d-c-c-b-a,（5）a-b-c-c-d-e-f-g,（6）h-j-a-b-c-c-d-e. The 6 trace pairs were sent to the subjects in different order by email. According to the judgments, the subjects choose one of results: identical (1.0), very similar (0.8), similar (0.6), different (0.4), very different (0.2), and absolutely different (0.0).

In addition, the experiment was published onto the website http://www.wenjuan.com/s/3EJzAb, so more volunteers can participate in the experiment. The results are illustrated as figure 2. The similarity between the target action trace and candidate action traces is computed and the result is illustrated as left figure in Fig. 2. The similarity between the target action trace and candidate action traces is manually estimated and the result is illustrated as right figure in Fig. 2.



(1) Computational Results (2) Manual Estimations

Fig. 2 the result of experiment

Conclusion

In this paper a new approach is proposed to measure similarity between action traces. Not only the sequence similarity but also the semantic heterogeneity is considered in this paper. The approach is more adaptive to the real application scenarios, in which the action is described by a textual message. So far, the work in this paper has been applies into a project of cross organization ERP implementation. In the future, the method still needs more projects to verify.

Acknowledgments

The research is supported by the Education Department of Heilongjiang province science and technology research projects (No. 1253G014).

The research is also supported by the Northeast Petroleum University youth science and technology research projects (No. 2013NQ118).

Reference

[1] Gerke, K., Cardoso, J.,Claus, A.: Measuring the compliance of processes with reference models. In: On the Move to Meaningful Internet Systems – Confederated International Conferences 2009, Proceedings, Part I, Springer(2009) :76-93

[2] LI Yan, FENG Yu-qiang. A Quantitative Analysis Method of Business Process based on Sequence Alignment. System Engineering Theory and Practice,2007, 27(4):54-61

[3] Haiping Zha, Jianmin Wang, Lijie Wen, Chaokun Wang, Jiaguang Sun . A workflow net similarity measure based on transition adjacency relations. Computers in Industry.2010, 61 (5) :463-471

[4] David Maier (1978). "The Complexity of Some Problems on Subsequences and Supersequences". J. ACM (ACM Press) 25 (2): 322–336

[5] Dekang Lin. An Information-Theoretic Definition of Similarity. In Proc. 15th International Conf. on Machine Learning (1998), pp. 296-304

[6] P. Bouguet, M. Ehrig, J. Euzenat, E. Franconi, P. Hitzler, M. Krotzscn, L. Serafini, G. Stamou, Y. Sure, and S. Tessaris, "Specification of A common framework for characterizing alignment," Knowledge Web Consortium 2005.

[7] P. Pantel and D. Lin, "Discovering Word Senses from Text," presented at Proceedings of ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2002.

[8] Harold W. Kuhn. "The Hungarian Method for the assignment problem,"Naval Research Logistics Quarterly, 2: 83-97, 1995

[9] Bergroth, L., Hakonen, H., Raita, T.: A survey of longest common subsequence algorithm. String Processing and Information Retrieval, International Symposiumon (2000). 39-48

[10] Smith, Temple F.; and Waterman, Michael S.. "Identification of Common Molecular Subsequences". Journal of Molecular Biology(1981) 147: 195–197.

Juntao GAO, received the PhD. degrees in Computer Science from BeiHang University in 2009. During 2009-2014, he stayed in Northeast Petroleum University to teach software engineering. His interest and research areas include process modeling, software requiremen-ts, semantic computing

Danchuan ZHAO, master student, majored in software engineering from Northeast Petroleum University. Her interest and research areas include software requirements, artificial intelegence.

Hongbo ZHOU, master, he is now works in the Northeast Petroleum University. His interest and research areas include information retrieval, clustering, and data integration.

Yongan WANG, master, he is now works in the Northeast Petroleum University. His interest and research areas include information retrieval, clustering, and data integration.