Detecting Concept Drift in Resource-Service Sequence for Collaborative Task

Tangquan Lin¹, Haibo Li^{2,*}, and Yingchuan Sun³

¹ College of Computer Science and Technology, Huaqiao University, Xiamen, China

² College of Computer Science and Technology, Huaqiao University, Xiamen, China

³ College of Computer Science and Technology, Huaqiao University, Xiamen, China

Abstract. To discover the bottleneck of business caused by the changing of market, especially in collaborative tasks, an approach is proposed to detect concept drift in the usage of resource services in business processes. Firstly, influence degree, described as features, is measured in a Resource-Service Sequence (RSS). Secondly, by mining business dataset, the influence relationship between resource services is resolved according to different time window. Then, the influence degrees are clustered as different clusters, called concept drift. Finally, RSSs with concept drift are derived from the feature sequence. The simulation results show the validity of the proposed approach.

1 Introduction

The development of techniques in cloud computing and Internet of Things results in gradually some changes of the modes in business management and work, which turns activities with single organization into collaborative tasks with multi-organization [1]. In collaborative task, workflow management system is used to integrate enterprise business, information, and resources to accomplish common tasks. Taking manufacturing as an example, the manufacturing process is no longer a single enterprise-owned operation in collaborative manufacturing mode, but participated by many enterprises, including product design companies, component suppliers, and sales companies, etc. In a collaborative task system, resources, as service, are published into a public cloud platform so that they are invoked conveniently in the execution of business activities, which are called resource service. With execution of business activities, there are temporal relationships between the resource services which are used by business activities. The resource services with temporal relationships are called Resource-Service Sequence (RSS). It is important to discover the set of RSSs to optimize the selection of service resources.

The current research on the relationship between resource services focuses on the following areas. The method based on parameter matching is one of important approaches, on which a relationship between services is built through matching their input and output parameters. Based on the relationship, a service-network model is established to provide more customized services for users [2]; A very different approach to the above is to consider dependency relationship depended by the functional semantics of web services. The dependency relationship can be organized into AND / OR graphs and applied to

service composition and so on [3]; Actually, with the development of semantic technology, semantic-based relationship among services has become a hot issue [4]. As described above, the current approaches to establish service-relationship models mainly include service parameters and semantic. However, these models do not describe the dynamic nature of service relationships. In fact, the use of resource services can be reflected by some indicators, called feature. The changing feature values can be directly mapped into the influence degree between resource services in an RSS. Therefore, by analyzing business data, the changing of influence degree between upstream and downstream resource services in an RSS can be detected. That is to say, concept drift [5] arises in the RSS. Detecting concept drift in RSS can help enterprises to discover bottlenecks of inter- organizational business process.

This paper is organized as follows. In Section 2, we analyze the problem about the concept drift. A mathematical description of the problem is presented. Section 3 describes the method in detail. The design of the experiments, and the results are presented in Section 4. Finally, in Section 5, we present a summary and discuss what we intend to do in future work.

2 Problem analysis

In a collaborative task system, there are decentralized resources provided by multi-organization and consolidated on a unified platform, to achieve shared tasks [6-7]. The platform utilizes a workflow management system to coordinate effectively the execution of business activities in a task, while the distributed resource-services are invoked by the workflow system to serve the business

* Corresponding author: lihaibo@hqu.edu.cn

activities in an orderly manner to ensure that the business process is successfully completed.

As shown in Fig.1, in the collaborative task platform, the process is shared by multiple organizations, because of the concentration of software resources, hardware resources, human resources and technical resources. The relevant definition about RSS can be given as follows.

Definition 1 Resource-service sequence Formally, let $RSS = \{ \langle R_1, R_2, \dots, R_n \rangle \mid n \leq |Act_{set}| \}$, where $R_1, \dots, R_n$ are apart sets of resource services invoked by activities in a activity set Act_{set} [8]. For example, there exist four resource-service sets R_1, R_2, R_3, R_4 , provided severally to $a_1 < a_2 < a_3 < a_4$, where $<$ indicates timing relationships between activities. We use $\langle \{r_1, r_2\}, \{r_3, r_4\}, \{r_3, r_5\}, \{r_3, r_6\} \rangle$ to denote for above RSS. If it is unfold by personal, other sub-sequences are got.

Fig. 1. The collaborative design and manufacturing process

Definition 2 Feature sequence Each resource service always has some features objectively. Moreover, there are also timing relationships between the features of adjacent resource services in an RSS. Thence, for $\langle r_1, r_2, \dots, r_n \rangle$, its Feature-Sequence [9] set is denoted by $FS_{set} = \{ \langle f_{1,j_1}, f_{2,j_2}, \dots, f_{l,j_l} \rangle \mid l \leq n \}$, where f_{l,j_l} represents any feature of a resource service r_l in $\langle r_1, r_2, \dots, r_n \rangle$.

Definition 3 Feature sequence with concept drift (FScd) Suppose $w_1, w_2, \dots, w_n$ representing a series of time windows, and the influence degrees between features $f_{n-1,j_{n-1}}$ and f_{n,j_n} at w_{n-1}, w_n are separately expressed as $I_{w_{n-1}}(f_{n-1,j_{n-1}} \rightarrow f_{n,j_n})$, $I_{w_n}(f_{n-1,j_{n-1}} \rightarrow f_{n,j_n})$. If $I_{w_n}(f_{n-1,j_{n-1}} \rightarrow f_{n,j_n})$ and $I_{w_{n-1}}(f_{n-1,j_{n-1}} \rightarrow f_{n,j_n})$ are classified respectively to two different concepts as which each cluster is treated, there is a concept drift to $I_{w_n}(f_{n-1,j_{n-1}} \rightarrow f_{n,j_n})$. The FScd consisting of two features is expressed as $\langle f_{n-1,j_{n-1}}, f_{n,j_n} \rangle$.

Definition 4 Resource-service sequence with concept drift (RSScd) For $\langle r_1, r_2, \dots, r_n \rangle$, if its each sub-sequence $\langle r_{n-1}, r_n \rangle$ consisting of two adjacent resource services all have at least a FScd $\langle f_{n-1,j_{n-1}}, f_{n,j_n} \rangle$, $\langle r_1, r_2, \dots, r_n \rangle$ is regarded as RSScd.

3 Algorithm statement

The approach to Detect Concept Drift in Resource-Service Sequences (**DCDRSS**) is introduced newly. It is generally detected that whether the influence degrees from upstream resources to downstream resources changes for an RSS.

3.1 Calculating the influence degree

To compute the influence degrees on features between upstream resources and downstream resources for an RSS, change rate is regarded as the measure of the influence degree. In the fields of mathematics application, change rate values the dependent variable as the independent variable changes. There are the larger the absolute values of the change rate, the greater the effect of the independent variable on the dependent variable, and vice versa. Taking the complete workflow execution process as a unit, the influence degree between feature $f_{n-1,j_{n-1}}$ and f_{n,j_n} in the process is defined as formula (1).

$$I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t)) = \frac{|f_{n,j_n}(t) - f_{n,j_n}(t-1)|}{|f_{n-1,j_{n-1}}(t) - f_{n-1,j_{n-1}}(t-1)|} \quad (1)$$

where $f_{n-1,j_{n-1}}(t)$ and $f_{n,j_n}(t)$ respectively are the values of the feature $f_{n-1,j_{n-1}}$ and f_{n,j_n} at time t . In the business data, the features of resource services are always variables over time. In other words, the value of every specific feature can form a time series in the course of workflow execution with a long execution cycle. Consequently, what we can get is that $I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t))$ also form a series of influence degrees.

Normalization is an important data preprocessing step in the field of data analysis. Considering that different resource-service features often have different measures, detecting results be greatly affected. In order to eliminate this impact, it is necessary to standardize the feature data. Here, we can adopt the following formula for unifying metric.

$$f_{n,j_n}^*(t) = \frac{f_{n,j_n}(t) - \min\{f_{n,j_n}(t)\}}{\max\{f_{n,j_n}(t)\} - \min\{f_{n,j_n}(t)\}} \quad (2)$$

The formula (2) maps all the values of resource-service features between $[-1, 1]$ through a linear transformation.

3.2 Clustering influence degrees

Given a series of influence degrees $\{I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t)) \mid t = 1, 2, \dots, T\}$, it is divided into several categories by clustering and each category represents a concept. Then whether the influence degrees $I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t))$ and $I(f_{n-1,j_{n-1}}(t-1) \rightarrow f_{n,j_n}(t-1))$ are classified to the same concept, is detected. If that isn't successful, there is a concept drift in $\langle f_{n-1,j_{n-1}}, f_{n,j_n} \rangle$ at t .

The following clustering:

I. The interval $[\min\{f_{n,j_n}(t)\}, \max\{f_{n,j_n}(t)\}]$ is equally divided into s intervals, respectively as $Q_m, m=1, 2, \dots, s$.

II. The dot density of $f_{n,j_n}(t)$ in $Q_m, f_{n,j_n}(t) \in Q_m$ is calculated as follows.

$$Density(f_{n,j_n}(t)) = \frac{range}{\sum |I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t)) - I(f_{n-1,j_{n-1}}(k) \rightarrow f_{n,j_n}(k))|} \quad (3)$$

where *range* is the range of each interval Q_m , $f_{n,j_n}(k) \in Q_m$, and $t \neq k$. The average of $Density(f_{n,j_n}(t))$ is equal to:

$$Density(Q_m) = \frac{\sum Density(f_{n,j_n}(t))}{num}, f_{n,j_n}(t) \in Q_m \quad (4)$$

where *num* is the number of influence degrees in Q_m .

III. Amalgamating Q_m by following rules: If $|Density(Q_{m-1}) - Density(Q_m)| < Density(C_z)$, where $Q_{m-1} \in C_z$, Q_m is merged to the cluster C_z ; The influence degrees in the intervals with sparse density require to be determine whether it can be amalgamated to the nearest cluster, or as discrete values.

IV. The cluster center is the value of influence degree with the highest dot- density value in the cluster.

The value of parameter *s* in the clustering algorithm depends on the range of $I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t))$ and its distribution. The smaller the range of $I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t))$ and the more concentrated its distribution is, the smaller *m* is.

3.3 Detecting concept drift in RSS

In collaborative task, there are always resource services with many features. It is more challenging to calculate the influence degree between the resource-service features and to detect concept drift in an RSS.

For simplicity, suppose there is the RSS $\langle r_1, r_2 \rangle$ consisting of r_1, r_2 , corresponding to the feature sets F_1, F_2 . The set comprised of the ordered pairs formed from the two features, one in F_1 as the first element, the other in F_2 as the second element, is called **Cartesian Product** [10], which is written as $C_1 = F_1 \times F_2$. For example, let $F_1 = \{f_{1,1}, f_{1,2}\}$ and $F_2 = \{f_{2,1}, f_{2,2}\}$, then $C_1 = \{\langle f_{1,1}, f_{2,1} \rangle, \langle f_{1,1}, f_{2,2} \rangle, \langle f_{1,2}, f_{2,1} \rangle, \langle f_{1,2}, f_{2,2} \rangle\}$. It can be observed that C_1 is also a set of feature sequences for $\langle r_1, r_2 \rangle$. Next, the each component for C_1 is detected whether a concept drift arises. Furthermore, the change of influence degree in feature sequence is calculated by formula (5).

$$\Delta I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t)) = I^*(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t)) - I^*(f_{n-1,j_{n-1}}(t-1) \rightarrow f_{n,j_n}(t-1)) \quad (5)$$

where $I^*(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t))$ is the center of the cluster to which $I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t))$ is categorized. So the changing of influence degree between r_{n-1} and r_n is measured as follows.

$$\Delta I_{r_{n-1}, r_n}(t) = \frac{\sum |\Delta I(f_{n-1,j_{n-1}}(t) \rightarrow f_{n,j_n}(t))|}{M} \quad (6)$$

where *M* is the number of FScd for $\langle r_{n-1}, r_n \rangle$.

Complicatedly, for $\langle r_1, r_2, \dots, r_n \rangle$, it is split into a binary sequence set $\{\langle r_{i-1}, r_i \rangle | i=2, 3, \dots, n\}$. It is similarly detected whether concept drift arises for each $\langle r_{i-1}, r_i \rangle$, $i=2, 3, \dots, n$. Finally, the set $RSScd_{set} = \{\langle r_{k-1}, r_k \rangle | k \in N\}$, where $N = \{2, 3, \dots, n\}$, composed of binary RSScd, is entered into **Algorithm RSScd_Splice** to get other RSScds.

Algorithm RSScd_Splice

Input: $RSScd_{set}$

Output: All RSScds.

```

1.   $r_1 \leftarrow A[m]$ 
2.   $r_2 \leftarrow B[m]$ 
3.   $next \leftarrow 0$ 
4.  Definding a string  $st$ 
5.   $st \leftarrow r_1[1]$  add to  $st$ 
6.   $next++$ 
7.  for  $i \leftarrow 1$  to  $num$ 
8.     $s \leftarrow r_2[i]$ 
9.     $positions[] \leftarrow k$ , subject to  $s == r_1[k]$ 
10.   if  $positions \neq \emptyset$  then
11.      $st \leftarrow r_2[i]$  add to  $st$ 
12.      $next++$ 
13.     for  $j \leftarrow 1$  to  $length(positions)$ 
14.        $s \leftarrow r_2[positions[j]]$ 
15.        $FS \leftarrow Find\_Seq(r_1, r_2, s, num, st)$ 
16.     end for
17.   else
18.     if  $num == 2$  then
19.        $st \leftarrow r_2[i]$  add to  $st$ 
20.        $FS \leftarrow st$  add to  $FS$ 
21.     end if
22.   end if
23. end for
24. return  $FS$ 

```

Procedure Find_Seq(r_1, r_2, s, num, st)

```

1.   $positions[] \leftarrow k$ , subject to  $s == r_1[k]$ 
2.  if  $positions \neq \emptyset$  then
3.     $st \leftarrow r_2[i]$  add to  $st$ 
4.     $next++$ 
5.    for  $j \leftarrow 1$  to  $length(positions)$ 
6.       $s \leftarrow r_2[positions[j]]$ 
7.       $FS \leftarrow Find\_Seq(r_1, r_2, s, num, st)$ 
8.    end for
9.  else
10.    $st \leftarrow s$  add to  $st$ 
11.    $FS \leftarrow st$  add to  $FS$ 
12. end if

```

4 Experiments and results

The collaborative design and manufacturing process of electronic products, as an example to illustrate the effectiveness of the proposed approach, include the whole business processes such as the design, manufacture, assembly and parts supply of electronic products, as shown in Fig.1. In the whole business process, product design includes hardware design, software design and mechanical structure design. The resources used in product design have mainly human resources and technical resources. Product processing largely consists of hardware production and machining. Hardware production,

machining and parts procurement employing hardware resources, human resources and technical resources, supply raw materials for product assembly. Business activities related to the collaborative design and manufacturing process of electronic products, invoke all resource services shown in Table 1. The features of each resource service are shown in Table 2.

Table 1. The resource-service categories for each business activity

Business activities and its resource service categories	Business activities and its resource service categories
Product design a_1 Product project book r_1 Overall designer r_2	Parts procurement a_8 Procurement scheme r_{15} Procurement officers r_{16}
Hardware design a_2 Overall design scheme r_3 Hardware designer r_4	Component configuration a_9 Part configuration scheme r_{10}
Software design a_3 Overall design scheme r_3 Software designer r_5	Machining a_{10} Mechanical structure design r_{11}
Mechanical design a_4 Overall design scheme r_3 Mechanical designer r_6	Mechanical processing equipment r_{12} Parts r_{13}
Hardware and software debugging a_5 Test engineer r_7 Hardware design scheme r_8 Software design scheme r_9	Product assembly a_{11} Parts r_{13} Mechanical components r_{17} Hardware r_{18} Product assembly equipment r_{19}
Part configuration a_6 Part configuration scheme r_{10}	Product testing a_{12} Product r_{20} Inspector r_{21}
Hardware production a_7 Software design scheme r_9 Parts r_{13} Hardware production equipment r_{14}	

Based on the workflow model shown in Fig.1, the simulation data is generated according to the following rule: Workflow logs generated, according to the executing frequency of business activities, are identical to the data created by assigning corresponding weights to five execution paths. For comparison, five paths are executed respectively according to the weights of $A = (0.1, 0.2, 0.2, 0.1, 0.4)$ and $B = (0.2, 0.2, 0.2, 0.2, 0.2)$ in the experimentation. Afterwards, **SPADE** [11] algorithm is adopted in the experimentation to mine the frequent RSSs. After being tested, RSScds are found in each frequent RSS. Limitedly, taking $\langle r_1, r_3, r_{10}, r_{19}, r_{21} \rangle$ as an example, the detecting of the results are shown in Table 3 and 4.

In Table 3 and 4, the influence degree between product project book and the overall design scheme changes 5.26, because the influence degree between the cost of product

project book and overall design scheme, the credibility of the overall design scheme respectively changes 8.43, 2.08. Others are similar. Running **Algorithm RSSCD_Splice**, the other RSScd, $\langle r_1, r_3, r_{10} \rangle$, is obtained.

Table 2. The features for each resource service

Resource-service categories and features	Resource-service categories and features
Product project book r_1 Cost $f_{1,1}$, Response period $f_{1,2}$	Mechanical processing equipment r_{12} Cost $f_{12,1}$, Response period $f_{12,2}$, Depreciation rate $f_{12,3}$
Overall designer r_2 Design fee $f_{2,1}$, Response period $f_{2,2}$, Satisfaction $f_{2,3}$	Parts r_{13} Cycle $f_{13,3}$, Cycle in transit $f_{13,4}$
Overall design scheme r_3 Cost $f_{3,1}$, Response period $f_{3,2}$, Credibility $f_{3,3}$	Hardware production equipment r_{14} Cost $f_{14,1}$, Response period $f_{14,2}$, Depreciation rate $f_{14,3}$
Hardware designer r_4 Design fee $f_{4,1}$, Response period $f_{4,2}$, Satisfaction $f_{4,3}$	Procurement scheme r_{15} Cost $f_{15,1}$, Response period $f_{15,2}$, Credibility $f_{15,3}$
Software designer r_5 Design fee $f_{5,1}$, Response period $f_{5,2}$, Satisfaction $f_{5,3}$	Procurement officers r_{16} Purchase fee $f_{16,1}$, Response period $f_{16,2}$, Satisfaction $f_{16,3}$
Mechanical designer r_6 Design fee $f_{6,1}$, Response period $f_{6,2}$, Satisfaction $f_{6,3}$	Mechanical components r_{17} Cost $f_{17,1}$, Response period $f_{17,2}$, Pass rate $f_{17,3}$
Test engineer r_7 Test fee $f_{7,1}$, Response period $f_{7,2}$, Test cycle $f_{7,3}$, Satisfaction $f_{7,4}$	Hardware r_{18} Cost $f_{18,1}$, Response period $f_{18,2}$, Pass rate $f_{18,3}$
Hardware design scheme r_8 Cost $f_{8,1}$, Response period $f_{8,2}$, Credibility $f_{8,3}$	Product assembly equipment r_{19} Cost $f_{19,1}$, Response period $f_{19,2}$, Depreciation rate $f_{19,3}$
Software design scheme r_9 Cost $f_{9,1}$, Response period $f_{9,2}$, Credibility $f_{9,3}$	Product r_{20} Cost $f_{20,1}$, Response period $f_{20,2}$, Assembly cycle $f_{20,3}$, Pass rate $f_{20,4}$
Component configuration scheme r_{10} Cost $f_{10,1}$, Response period $f_{10,2}$, Credibility $f_{10,3}$	Inspector r_{21} Inspection fee $f_{21,1}$, Response period $f_{21,2}$, Detection cycle $f_{21,3}$, Satisfaction $f_{21,4}$
Mechanical structure design r_{11} Cost $f_{11,1}$, Response period $f_{11,2}$, Credibility $f_{11,3}$	

Using the same data, detecting results of the concept drift are compared, adopting algorithm in the paper and **DivideSlope** in [9], as are shown in Fig. 2. It presents that

the number of RSScds detected by the algorithm proposed in this paper is significantly more than **DivideSlope**. The main advantage of the algorithm proposed in the paper is the fact that the influence degrees of changing are still kept and are categorized apart into different cluster, when the changing of influence degrees in value is large in certain time windows. Conversely, values of influence degrees in algorithm **DivideSlope** are the average of the steady changing rate. Due to the neglect of non-stationary changing rate, the most of influence degrees change a little in calculation over time and are categorized to the same concept.

Table 3. FScds consisting of two features

Feature sequences	ΔI	Feature sequences	ΔI
$\langle f_{1,1}, f_{3,1} \rangle$	-8.43	$\langle f_{3,3}, f_{10,2} \rangle$	-1.93
$\langle f_{1,1}, f_{3,3} \rangle$	2.08	$\langle f_{19,1}, f_{21,1} \rangle$	2.14
$\langle f_{3,1}, f_{10,1} \rangle$	13.69	$\langle f_{19,2}, f_{21,1} \rangle$	3.36
$\langle f_{3,2}, f_{10,1} \rangle$	-7.26	$\langle f_{19,3}, f_{21,1} \rangle$	2.11
$\langle f_{3,3}, f_{10,1} \rangle$	-12.95		

Table 4. RSScds consisting of two resource services

Resources-service sequences	$\Delta I(r_{n-1} \rightarrow r_n)$
$\langle r_1, r_3 \rangle$	5.26
$\langle r_3, r_{10} \rangle$	8.96
$\langle r_{19}, r_{21} \rangle$	2.54

Fig. 2. Comparison of experimental results for **DCDRSS** and **DivideSlope**.

5 Conclusion and future work

We propose a detecting approach to conceptual drift of resource-service sequences depending on the influence relationship between features for collaborative task. Through computing the changing rate between features in different time window, to construct a time series of influence degrees, the values of influence degree are classified by a clustering algorithm, and each cluster forms a concept. According to the obtained concept, it is detected whether the conceptual drift arose in an RSS. Finally, it is verified by an experiment that the approach can detect concept drifting well for an RSS. In the future, we intend to investigate other approaches for two-way influence relationship between resource services.

Acknowledgments

This work was supported in part by a grant from 71571056 of the Natural Science Foundation of China (NSFC), JZ160409 of the Natural Foundation Key Program for Young Scholars in the Universities of Fujian Province, the Scientific Research Funds of Huaqiao University (16BS304) and the Central Universities (Grant No. JB-ZR1147).

References

1. Q. Mo, F. Dai, R. Zhu, J. Da, et al. Comput. Integra. Manuf. Sys. **21**,1052(2015)
2. G. Q. LIU, Z. WANG, J. F. REN, et al. J. Northeast. Univ. **34**,198(2013)
3. D. H. Shin, H. B. Jeon, and K. H. Lee.2010 IEEE Asia-Pacific.**8**,168(2010)
4. Q. Liang, S. Chen, and Z. Feng. Appl. Math. Inform. SCI.**7**,245(2013)
5. A. Tsymbal. Technical report, Department of Computer Science Trinity College.**2**,451(2004)
6. B.LI, X. CHAI.ZTE Commu.**16**,7(2010)
7. S. ZHANG, and H. B. LI. Comput. Integra . Manuf. Sys.**21**,836(2015)
8. H. B. LI, and M.X. LIAN, Comput. Integra . Manuf. Sys. **22**,1791(2016)
9. H. B. LI, and X. Y. LEI. Comput. Integra . Manuf. Sys.**23**,2579(2017)
10. H. T. Huang, D. Y. Bi, Y. F. Zha, et al. J. Electron. Inf. Techno.**37**,518(2015)
11. J. Zaki. Mohammed.Mach.Learn. **42**,51(2001)