Tsinghua Science and Technology

Volume 22 | Issue 2 Article 5

2017

Similarity Search Algorithm over Data Supply Chain Based on Key Points

Peng Li
School of Computer Science, Beijing University of Posts and Telecommunication, Beijing 100876, China.

Hong Luo
School of Computer Science, Beijing University of Posts and Telecommunication, Beijing 100876, China.

Yan Sun
School of Computer Science, Beijing University of Posts and Telecommunication, Beijing 100876, China.

Follow this and additional works at: https://tsinghuauniversitypress.researchcommons.org/tsinghua-science-and-technology

 Part of the Computer Sciences Commons, and the Electrical and Computer Engineering Commons

Recommended Citation

Peng Li, Hong Luo, Yan Sun. Similarity Search Algorithm over Data Supply Chain Based on Key Points. Tsinghua Science and Technology 2017, 22(2): 174-184.

This Research Article is brought to you for free and open access by Tsinghua University Press: Journals Publishing. It has been accepted for inclusion in Tsinghua Science and Technology by an authorized editor of Tsinghua University Press: Journals Publishing.

TSINGHUA SCIENCE AND TECHNOLOGY
ISSN 1007-0214 05/09 pp174–184
Volume 22, Number 2, April 2017

Similarity Search Algorithm over Data Supply Chain Based on Key Points

Peng Li, Hong Luo*, and Yan Sun

Abstract: In this paper, we target a similarity search among data supply chains, which plays an essential role in optimizing the supply chain and extending its value. This problem is very challenging for application-oriented data supply chains because the high complexity of the data supply chain makes the computation of similarity extremely complex and inefficient. In this paper, we propose a feature space representation model based on key points, which can extract the key features from the subsequences of the original data supply chain and simplify it into a feature vector form. Then, we formulate the similarity computation of the subsequences based on the multiscale features. Further, we propose an improved hierarchical clustering algorithm for a similarity search over the data supply chains. The main idea is to separate the subsequences into disjoint groups such that each group meets one specific clustering criteria; thus, the cluster containing the query object is the similarity search result. The experimental results show that the proposed approach is both effective and efficient for data supply chain retrieval.

Key words: data supply chain; similarity search; feature space; hierarchical clustering

1 Introduction

The amount of data generated has been continuously growing from global data sources such as websites, social media, mobile applications, news networks, weather, political institutes, societies, and the economy[1]. Large amounts of data have been collected and are widely available on the data platforms. Besides, this platform facilitates the creation, validation, and execution of the data analysis algorithm[2]. Data platforms enable data to flow freely for the benefit of whole organizations. Organizations will be able to exploit big data for competitive advantages. A data supply chain is constructed when data is created, transformed, combined with other data, and exported to the next user[3]. The data then moves, flows, and transforms through the supply chain, incrementally acquiring value. Data supply chains can help break down the boundaries of an enterprise. As a result, organizations have the opportunity to ingest new sources of data for their business. Significant efforts have been made in developing novel similarity search algorithms among data supply chains due to their promising applications. For example, a similarity query identifies those data supply chains whose structure evolved similar to a specific one. It is not only offering users the best candidates of data supply chains to optimize their products, but also helps find the potential consumers of their data and extends its value. In this paper, we target the problem of providing a similarity search among data supply chains at high precision and efficiency to meet the needs of applications.

Cluster analysis[4,5] is an important technique in data mining and data analysis, so it can be used in a similarity search of a data supply chain. However, the performance of the cluster-based similarity search is severely affected due to the following reasons. First, many clustering approaches seldom differentiate between global similarity and local similarity of data

• Peng Li, Hong Luo, and Yan Sun are with School of Computer Science, Beijing University of Posts and Telecommunication, Beijing 100876, China. E-mail: {luoh, sunyan}@bupt.edu.cn.
• Hong Luo and Yan Sun are also with the Beijing Key Lab of Intelligent Telecommunication Software and Multimedia, Beijing 100876, China.
* To whom correspondence should be addressed.
Manuscript received: 2016-11-25; revised: 2016-12-21; accepted: 2017-01-03

supply chains, which reduces the processing rate and clustering quality. Second, if using the existing clustering algorithms to cluster the original data supply chains, the efficiency degrades rapidly with the increase of the number of nodes.

In this paper, we design a Similarity Search System for Data Supply Chain (SSS-DSC), which searches for the most similar object (data supply chain) to a given one. Developing a practical system from basic principles, however, entails substantial challenges. First, when the reference data supply chains are extremely large, the efficiency of the similarity search is affected by the number of nodes. How to abstractly represent the original data supply chains and retain the intrinsic feature for improving the searching efficiency remains challenging. Second, to the best of our knowledge, there are seldom reported approaches that can calculate a distance measure for measuring the closeness of the corresponding unequal data supply chains and we need to choose a distance function and carefully sidestep the problem.

To tackle the above challenges, a novel feature space representation model based on key points is proposed. We first seek and extract the key points. Using these key points, the original data supply chains can be partitioned into several subsequences (subchains). Then, we extract the feature of each subsequence and construct a feature space to represent the original DSC. To tackle the previously low precision of a distance measure for unequal data supply chains, we further develop a novel similarity computation algorithm with multidimensional features. Subsequences are characterized in multidimensional feature vector form. For features in different dimensions, we calculate the distances of each pair of subsequences by different distance formulas, and integrate different values with linear weights. Our algorithm reaches the most similar results according to specific criteria, which performs subsequence matching and subsequence searching. Subsequence searching means that the query pattern may be comprised between any nodes in the candidate sequence. The main contributions of this paper are threefold.

(1) We design and implement a similarity search system for data supply chains. To the best of our knowledge, SSS-DSC is the first system that can successfully achieve an efficient similarity search for data supply chains in data platforms. (2) We propose a novel feature space representation model based on key points, which represents the original data supply chains by applying a feature extraction technique and improves the quality and efficiency of the data supply chain similarity search. (3) We conduct simulation experiments, and the experimental results show that the proposed approach outperforms the existing algorithms by at least 20% in query performance.

The rest of the paper is organized as follows. We summarize the background and related work in Section 2. We formally define the problem in Section 3. We elaborate the design and implementation of our system framework in Section 4. In Section 5, we present the details of algorithms for representing the data supply chain and similarity measurement with multiple features, followed by the performance evaluation in Section 6. In Section 7, we conclude the paper.

2 Related Work

A similarity search for data supply chains is an important function in many applications, and has drawn much attention in recent decades. The report shows the similarity between data supply chains and time series. Previous researches on the similarity search have mainly focused on time series approximate representations, similarity measure methods, and time series clustering.

Cui et al.[6] developed a novel framework for an efficient similarity search on TSC data. The framework addressed the following issues. First, it provided a compact representation for the TSC data. Second, it used a multidimensional relation vector to capture the natural relations between the multiple time series in a TSC. However, the framework may lead to a loss of important information, and it does not work well for all time series. Due to its quadratic time and space complexity, DTW is not suitable for large time series datasets. Yin et al.[7] presented a novel parallel scheme for a fast similarity search based on DTW, which is called the MapReduce-based DTW. The experimental results showed that this approach not only retained the original accuracy as DTW, but also greatly improved the efficiency of the similarity measure in large time series. This technique worked well when all features had the same units of scale; however, it was often ineffective for combining disparate features. Iwashita et al.[8] proposed a method of determining the optimal number of clusters, at which the fastest retrieval performance could be

obtained. The problem was challenging because categorical variables make it difficult to define a meaningful distance between trajectories in clustering multivariate time series. Ghassempour et al.[9] proposed an approach based on Hidden Markov Models (HMMs), where they first mapped each trajectory into an HMM, then defined a suitable distance between HMMs and finally proceeded to cluster the HMMs with a method based on a distance matrix. However, this method did not consider any errors that were incurred. Rakthanmanon and Keogh[10] introduced a useful concept of time series shape-lets. In this work, they proposed a fast shape-lets discovery algorithm that could be used to classify unlabeled time series. Karamitopoulos and Evangelidis[11] presented a new method that accelerated the similarity search implemented via one-nearest neighbor on the time series data. The idea was to map the original data into a lower dimension domain without losing a substantial amount of information. The approach of dimensionality reduction can be very helpful because it reduces the storage requirements, it potentially allows an efficient implementation of multidimensional indexing structures and it improves the quality of the similarity search results.

Besides the techniques described above, a semantic-aware domain of works are related to our technique. A semantic web services based approach is developed in Ref. [12], where semantics is applied for supporting the semi-automatic process planning and chaining procedure. Meng et al.[13] proposed a Keyword-Aware Service Recommendation method, named KASR, to address the big data analysis problem for service recommender systems. In this paper, keywords are used to indicate the users' preferences, and a user-based collaborative filtering algorithm is adopted to generate the appropriate recommendations. To retrieve and recommend subchains of possible service invocations to a certain user, Zhou et al.[14] calculated the semantic similarity between operations of services through considering the semantic similarity of the name and text descriptions of operations, for better capturing the invocation possibility between operations.

To summarize, these techniques are inspiring to us for proposing the similarity search algorithm over data supply chains, based on key points.

3 Problem Definition

A data supply chain is treated as an object in this paper; it consists of plentiful dynamic time-series data. To provide a convenient expression, we give some definitions as follows.

Definition 1 (Data Supply Chain Set) A set of data supply chains, denoted by $\Sigma = \{S_1, S_2, ..., S_n\}$, where n is the serial number of data supply chain.

Definition 2 (Data Supply Chain) Given a data supply chain S, which consists of a data sequence ordered by the generation time. A data supply chain is denoted by $S = \{d_1, ..., d_{t_i}, ..., d_n\}$, where d_{t_i}, $t_0 < t_i < t_n$, is an instance of data generated at t_i.

Definition 3 (SubSequence) Given a data supply chain S of length n, a subsequence of S is a sampling of length m ($m \leqslant n$) of contiguous positions from S, that is, $\beta = \{d_{t_p}, ..., d_{t_{p+m-1}}\}$, $1 \leqslant p \leqslant n-m+1$.

Definition 4 (Data Node) A data node d of a data supply chain is a two-tuple, which is denoted by $d =$ (nm, descp), where nm is the name of this node, and descp is the text description of this node.

Definition 5 (Segment Feature) Consider a data supply chain S that has been segmented into k subsequences $\{\beta_1, \beta_2, ..., \beta_k\}$, SF_i is a triple of feature vector of the i-th subsequence β_i such that

$$\mathrm{SF}_i = (\mathbf{ARS}_i, \mathbf{AP}_i, \mathbf{DES}_i) \tag{1}$$

Here, $\mathbf{ARS}_i$ is the feature vector representing the association rules set of β_i; $\mathbf{AP}_i$ is the feature vector of the application purpose; and $\mathbf{DES}_i$ is the feature vector representing its evolution.

Definition 6 (Distance) Given two segment features SF_1 and SF_2 representing β_1 and β_2, respectively, the distance between β_1 and β_2 is given by

$$D(\beta_1, \beta_2) = w_1 \cdot d_1(\mathbf{ARS}_1, \mathbf{ARS}_2) + w_2 \cdot d_2(\mathbf{AP}_1, \mathbf{AP}_2) + w_3 \cdot d_3(\mathbf{DES}_1, \mathbf{DES}_2) \tag{2}$$

where $d_i()$ is the distance of each feature vector and w_i ($1 \leqslant i \leqslant 3$) is the weight associated with a specific attribute. The summation of all weights is 1.

Definition 7 (Similarity Calculation) Given a reference data supply chain or subsequence of chain Q and its segment feature SF_q, a set of data supply chains Σ, and a user specified distance threshold ε, a similarity search retrieves all data supply chains $S_i \in \Sigma$ such that

$$D(\mathrm{SF}_q, \mathrm{SF}_j) \leqslant \varepsilon \tag{3}$$

where $\varepsilon > 0$. If Formula (3) is established, it is stated that Q and subsequence β_j of S_i are similar to the case of the ε boundary.

The similarity search basic problem can be stated as follows: given a set of objects, find the most similar ones to a given query object.

4 System Framework

Figure 1 shows the framework of the similarity search system for data supply chains. As depicted in Fig. 1, the similarity search for data supply chains is provided according to the user's input. The similarity search process consists of three phases that are described hereafter:

(1) Feature exaction and modeling: this is the core of the system. Here, we propose a novel Feature Space Representation Model based on Key Points (FSRM-KP). FSRM-KP first seeks and extracts the key points for each data supply chain, then divides each chain into a set of subsequences using these points (also called boundary points). Then, several features can be extracted from the subsequences such as Association Rule Sets (ARS), Application Purpose (AP), and Data Evolution Sequence (DES). Thus, we construct a feature space for each subsequence and describe the original data supply chains according to the feature space model. Using this method, the storage of each chain is shrunk significantly.

(2) Similarity measure based on multidimensional features: we design a similarity measurement algorithm based on the feature space model. Feature spaces are divided into three feature classes: ARS, AP, and DES. By dividing the feature spaces into the above classes, we calculate distances of each pair of subsequence features using the available Natural Language Processing (NLP)

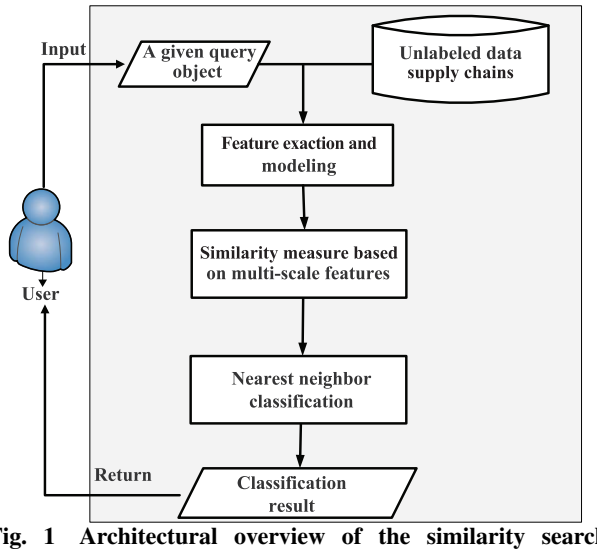


Fig. 1 Architectural overview of the similarity search system for data supply chains.

APIs and edit the distance techniques. Further, we get the pairwise distance of the subsequence by integrating different distance values with linear weights.

(3) Nearest neighbor classification: a hierarchical clustering algorithm for data supply chains is proposed. Since the proposed FSRM-KP presents features of subsequences, we choose those as a new specific clustering criteria. The proposed clustering algorithm processes the transformed subsequences and outputs the similarity search result.

5 Similarity Search for Data Supply Chains

5.1 A semantic-aware discovery method for key points

Typically, several subchains are chained together to construct a data supply chain. The specific function can be satisfied by a subchain, which consists of similar data nodes. In this setting, we leverage the semantic similarity between the name and text description of the data nodes to find the key points.

5.1.1 Semantic similarity between data node names

The name of a data node is normally specified as a phrase composed of several words. To express the name concisely, keywords extracted from the name are used to indicate its topic. In our method, two data structures, keyword-candidate list and specialized domain thesaurus, are introduced. The keyword-candidate list is a set of keywords about the name's topic, which can be denoted as $K = \{k_1, k_2, ..., k_n\}$, where n is the number of the keywords in the keyword-candidate list. The name's topic will be formalized into a keyword set. Usually, some words in names cannot exactly match the corresponding keywords in the keyword-candidate list, which characterizes the same aspects as the words. In this paper, we assume that the specialized domain thesauruses are built to support the keyword extraction, and different domain thesauruses are built for different domains. A domain thesaurus is a reference work of the keyword-candidate list that lists words grouped together according to the similarity of the keyword meaning, including synonyms and contrasting words and antonyms[15].

Next, the topics of the active node and the successor node are formalized into their corresponding topic keyword sets. In this paper, an active node refers to a current node and a successor node refers to a successor

of the current node. The topic keyword set of the active node can be denoted as ANK = $\{\text{ank}_1, \text{ank}_2, ..., \text{ank}_l\}$, where ank_i, $1 \leqslant i \leqslant l$, is the i-th keyword extracted from the name and l is the number of extracted keywords. The topic keyword set of the successor node can be denoted as SNK = $\{\text{snk}_1, \text{snk}_2, ..., \text{snk}_h\}$, where snk_i, $1 \leqslant i \leqslant h$, is the i-th keyword extracted from name and h is the number of extracted keywords. The keyword extraction process is described as follows:

(1) **Preprocess** Stop words in the name should be removed to avoid affecting the quality of the keyword extraction in the next stage. Also, the Porter Stemmer algorithm (keyword stripping)[16] is used to remove the common morphological endings from words. Its main use is as part of a term normalization process.

(2) **Keyword extraction** In this phase, the node name will be transformed into a corresponding keyword set according to the keyword-candidate list and domain thesaurus. If the node name contains a word in the domain thesaurus, then the corresponding keyword should be extracted into the topic keyword set.

A frequently used method, the Jaccard coefficient, is applied in the similarity computation between the names of data nodes. The Jaccard coefficient is a measurement of asymmetric information on binary (and non-binary) variables, and it is useful when negative values provide no information. The similarity between the topics of the active node and a successor node, based on the Jaccard coefficient, is described as follows:

$$\text{SimNa}(\text{ADK}, \text{SDK}) = \frac{|\text{ADK} \bigcap \text{SDK}|}{|\text{ADK} \bigcup \text{SDK}|} \quad (4)$$

where ADK is the topic keyword set of an active node and SDK is the topic keyword set of the successor node.

5.1.2 Semantic similarity between data node text descriptions

In this section, we propose to calculate a similarity degree between the text descriptions of an active node and the successor node, which can be denoted as SimDescp (ADT, PDT), where ADT is the text description of an active node and PDT is the text description of the successor node. A text description includes one or several sentences. We adopt an algorithm named xsimilarity (https://github.com/iamxiatian/xsimilarity) to obtain the text similarity. This algorithm has been developed and made accessible as open source software. A degree of text similarity is computed through considering both the semantic similarity between the words in sentences (denoted WordSim) and the order of these words (denoted OrdSim). The similarity computation process is described as follows.

(1) Words in sentences are extracted and stored in a vector while keeping the sequence of these words. Given two vectors $\mathbf{vec}_1$ and $\mathbf{vec}_2$, the word similarity is computed in a pairwise fashion. The semantic similarity degree is recorded in a vector **VecWdSim**=<wd_1, wd_2, WdSim>, where wd_1 represents a word in $\mathbf{vec}_1$, wd_2 represents a word in $\mathbf{vec}_2$ and WdSim represents the semantic similarity degree between wd_1 and wd_2. The values of word similarity degrees are sorted in descending order, and we denote this sort by using SORT.

(2) We first compute the semantic similarity between the words of two vectors. The vector **VecWdSim** on top in SORT is retrieved and the variable WdSimTtl is calculated as follows:

$$\text{WdSimTtl} = \text{WdSimTtl} + \text{VecWdSim.WdSim} \quad (5)$$

when VecWdSim.WdSim is no less than a pre-specified threshold, **VecWdSim** is inserted into another list SORT_{ord}, which is used for computing the similarity of word orders in the following. We remove any vector from SORT when this vector contains either VecWdSim.wd_1 or VecWdSim.wd_2. We iterate this procedure until SORT is empty.

(3) We then compute the similarity considering the order of words in the text descriptions. When SORT_{ord} is an empty set, OrdSim is set to 0. Otherwise, we number the words in the vectors of SORT_{ord} according to their order in $\mathbf{vec}_1$ and $\mathbf{vec}_2$. Given two pairs of words in any two neighboring vectors in SORT_{ord}, if their numbers of order are reversed, we set the variable RevOrdCnt as RevOrdCnt = RevOrdCnt + 1. We study neighboring vectors pairwise in SORT_{ord} to explore all reverse ordering cases. Finally, based on the above descriptions, a similarity degree SimDescp between the text descriptions of the two nodes is computed.

5.1.3 A semantic-aware discovery algorithm for key points

After computing the semantic similarity for the name and text description of the data nodes, the similarity of two data nodes is computed through the following formula:

$$\text{SimNode}(\text{AD}, \text{SD}) = \delta \cdot \text{SimNa}(\text{ADK}, \text{SDK}) + (1 - \delta) \cdot \text{SimDescp}(\text{ADT}, \text{PDT}) \quad (6)$$

where AD represents an active node, SD represents a successor node, the parameter δ implies the

relative importance of SimNa(ADK, SDK) and SimDescp(ADT, PDT) for the similarity calculation. It is observed that the number of words contained in names is far fewer than that contained in a text description. This indicates that the text description should contribute more on distinguishing between two nodes than the name.

Based on the abovementioned, we propose a semantic-aware discovery algorithm for key points. The algorithm is to seek the key points for each data supply chain. Let Σ denote a set of data supply chains, ρ denote a pre-specified threshold, S denote a data supply chain, and $\mathrm{KP} = \{\mathrm{kp}_1, \mathrm{kp}_2, ..., \mathrm{kp}_m\}$ denote a set of key points. The semantic-aware discovery algorithm for key points is shown as Algorithm 1.

We assume there are K data supply chains and P average number of nodes for each data supply chain. The time complexity of Algorithm 1 is $O(KP)$, since one needs to iterate over all the nodes of each chain to discover the key points. Along with the scaling-up numbers of K and P, the discovery time displays an upward tendency.

5.2 Feature space representation model based on key points

To reduce the computation time and improve the search efficiency, the data supply chains must be reduced in complexity. Hence, we propose a feature space representation model based on key points. The basic idea of FSRM-KP provides the oscillation behavior of a data supply chain that has been transformed into a feature space by linear segments. This representation, however, depends on several points chosen in the segmentation process. Demonstrating a data supply chain using one feature may not be sufficient to describe the actual oscillation trends. To solve this, we extract several features from the subsequences such as association rules sets, application purpose, and data evolution sequence, and extend the solutions to a multidimensional approach. Each subsequence includes three feature vectors. We adopt the frequent pattern mining algorithm as the basic algorithm and add the temporal constraints to discover the correlation among multiple data nodes and obtain the association rules set. By adding the sequential constraint and the time factor, the algorithm achieves more precise mining and shorter computation. Using the standard provenance (PROV) technology, we get the attribute arguments, which depict the actions performed on the data and the entities being responsible for those actions. Each PROV record, which contains identity information, activity, time of occurrence, and consumer demand, is stored in the PROV database. Therefore, we can extract consumer purpose and a data evolution sequence from it. Data evolution sequence is composed of data and the operations associated with the data. Formally, a subsequence is defined as a tuple. Furthermore, a data supply chain is represented by a matrix consisting of n segments and three features.

Let $S \in \Sigma$ denote a data supply chain and SF denote the segment feature of the subsequence. The feature space model transforming algorithm based on key points is shown as Algorithm 2.

Algorithm 1 A Semantic-Aware Discovery Algorithm for Key Points

Input: S, Σ, ρ
Output: KP

```
KP ← ∅
for each S ∈ Σ do
    Cur_Node ← S.initial;
    repeat
        Act_Node ← Cur_Node;
        Succ_Node ← Cur_Node.successor;
        Sim_val ← SimNode(Act_Node, Succ_Node)
        if Sim_val< ρ then
            KP ← Cur_Node;
        end if
        Cur_Node ← Cur_Node.successor;
    until Cur_Node=S.ending;
    Act_Node ← Cur_Node;
end for
Return KP;
```

Algorithm 2 Feature Space Model Transforming Algorithm based on Key Points

Input: S
Output: $\mathrm{SF}_1, \mathrm{SF}_2, ..., \mathrm{SF}_n$ // n is the number of segments of all data supply chains

```
Seek and extract key points from S using Algorithm 1;
Segment S into n sections {β1, β2, ..., βn} using these key points;
for each subsequence ∈ S do
    Extract association rules set, application purpose and data evolution sequence from subsequence;
    Construct the feature space for subsequence SF = (ARS, P, DES);
end for
Return SF1, SF2, ... , SFn;
```

We assume there are n segments of S. When employing the FSRM-KP approach, the time complexity of Algorithm 2 is $O(n)$ since each subsequence is represented by three features.

5.3 Similarity measure based on multi-scale features

In the previous section, we demonstrated how to computationally reduce the complexity of a data supply chain, representing it by the major turning points and feature space. This transformation is obviously required for the searched candidate sequences. The similarity measure can efficiently support the similarity search, which directly influences the shape of the clusters; the next step is to define the distance function. The use of multidimensional features causes the problem of measuring the similarity between two data supply chains to become one of measuring the distance between the two data supply chains of the feature vector. For this reason, a suitable similarity measurement algorithm based on this should be given. The comparison between two data supply chains is performed in two basic steps. First, the data supply chains of features relative to each scale are compared, using the different distance function defined. The proposed FSRM-KP supports several kinds of distance functions, in our implementation, we distinguish features in different dimensions and those distances are usually measured by different distance formulas.

5.3.1 Similarity measurement method for association rules set

ARS is a set of association rules, which can describe the correlation among multiple data nodes of a region. It can be described as

$$\mathrm{ARS} = (\mathrm{AR}_1, \mathrm{AR}_2, ..., \mathrm{AR}_n) \tag{7}$$

where AR_i is an association rule with support S.

Let ARS_1 and ARS_2 denote different association rules sets, $\mathrm{ARS}_1 \neq \varnothing$, $\mathrm{ARS}_2 \neq \varnothing$, the distance between ARS_1 and ARS_2 is given by

$$d(\mathrm{ARS}_1, \mathrm{ARS}_2) = \frac{|\mathrm{ARS}_1 \bigcap \mathrm{ARS}_2|}{|\mathrm{ARS}_1 \bigcup \mathrm{ARS}_2|} \tag{8}$$

where $|\mathrm{ARS}|$ denotes the number of association rules set.

5.3.2 Similarity measurement method for application purposes

Comparing AP helps us to compute a more accurate similarity ranking. All AP attributes are text-based that include information such as consumer demand and the objective of the data analysis. According to its characteristics, the similarity measure task is performed through available NLP APIs. By using third party NLP APIs that add semantic annotations or tagging to data supply chain of texts, we can extract a topic/key word from each one. To perform this task, many potential NLP web APIs have been considered and tested. They include Wikimeta, OpenCalais, Pingar, AlchemyAPI, and Semantria[17]. In many cases the NLP service may not be able to return a correct topic name for a given text. To obtain a larger number of topic names, multiple NLP services are used in conjunction with each other. OpenCalais allows for 50 000 API calls a day and 4 calls per second as part of the free license. AlchemyAPI provides up to 30 000 API calls a day for research purposes. Once all application purpose features are established, we will attempt to find commonality among the obtained topics to compute the distance value between each subsequence and a given one.

5.3.3 Similarity measurement method for data evolution sequence

To determine the similarity of two data evolution sequences, an approximate symbol matching algorithm based on edit distance[18] is used. Its main idea is the following: the more similarity between two data evolution sequences, the less number of data transformation operations required to transform one data evolution sequence into the other. The data transformation operation can be weighted by an arbitrary weight function that assigns each data transformation operation a numeric value. The sequence distance is a numeric value that represents the sum weight of the data transformation operations, which is required to equalize two data evolution sequences. Let S and T denote two data evolution sequences, $O_{\mathrm{sum}} = \{O_1, O_2, ..., O_n\}$ denote a set of data transformation operations sequence transforming S into T, $t(O_i)$ denote a weight of data transformation operation. Given $T(O_{\mathrm{sum}}) = \sum_{i=1}^{k} t(O_i)$, the sequence distance $d(S, T)$ between S and T is then defined as

$$d(S, T) = \min\{T(O_{\mathrm{sum}}) | O_{\mathrm{sum}} \text{ is a set of transformations of } S \text{ into } T\} \tag{9}$$

In the final step, the different distance values are integrated with the linear weights. The weight assignment is based on the distance values. We assign

more weight for the smaller value of feature, which avoids each feature vector significantly affecting the final results.

5.4 An improved hierarchical clustering algorithm

Up to this point, the data supply chains are expressed in terms of the feature space model and the distance measure formula is defined. To provide more accurate results, we propose a hierarchical clustering algorithm for data supply chains, which differentiates between global similarity and local similarity of data supply chains and performs subsequence matching and subsequence searching. The algorithm can improve the efficiency while consecutively maintaining the accuracy. The basic idea of the algorithm is as follows: first, the original data supply chain is divided into a set of subsequences represented by feature model; then, each subsequence is called as a cluster. According to the abovementioned similarity measure approach, the distances between each cluster are measured. We separate subsequences into disjoint groups such that the same-group of subsequences meets a specific clustering criteria. The cluster, which the query object lies within, is the similarity search results.

Let Σ denote a set of data supply chains, Q denote a reference data supply chain or subsequence of a chain, C_i denote the i-th cluster, ε denote a user specified distance threshold, C_{results} denote the cluster including the query object of the subsequence and the most similar ones. An improved hierarchical clustering algorithm for data supply chains is shown as Algorithm 3.

Algorithm 3 A Hierarchical Clustering Algorithm for Data Supply Chains

Input: Q, Σ, ε
Output: C_{results}
1: **for** each S $\in \Sigma$ **do**
2: $\{\text{SF}_1, \text{SF}_2, ..., \text{SF}_n\} \leftarrow$ FSRM-KP(S);
3: $C_i \leftarrow \text{SF}_i$ // C_i indicates a cluster ;
4: **end for**
5: **repeat**
6: Compute the distances between each pair of clusters by using the similarity measure approach;
7: Find the most similar clusters C_i and C_j , where C_i and C_j come from different data supply chains;
8: Merge them into one cluster and update the center of the generated cluster;
9: **until** the distances between each pair of clusters are beyond the ε specified by the user
10: Return C_{results};

The time complexity of Algorithm 3 is $O(Kn + K^2n^2)$. The query time is directly affected by K and n since the approach needs to iterate over all clusters.

6 Experimental Analysis

6.1 Experimental setup

In this section, experiments are designed and analyzed. The proposed algorithms are implemented with the Java programming language. The experiments are conducted on a computer with an Intel(R) Core(TM) i5-3337U processor, 4 GB memory, and the 64-bit Windows 7 system. To the best of our knowledge, there are seldom authoritative datasets and reported approaches that provide a similarity search for data supply chains. Hence, the experiments are conducted on synthetic datasets to evaluate the performance of the proposed approach. We have built a data platform. Users can publish and download data in this platform. The datasets are generated after simulation users upload and download data multiple times. All data supply chains are labeled according to the class they belong to. The number of classes is 10 in the datasets.

To evaluate the performance of a Semantic-Aware Discovery Algorithm for Key Points (SADA-KP) in accuracy, we compare SADA-KP with two other methods: Trend-Based Similarity Search in Time-Series Data (TBSS-TSD)[19] and Shape-based Similarity Measure for Time Series Data with Ensemble Learning (SSH-TSDEL)[20]. To evaluate the query accuracy and time of a Hierarchical Clustering Algorithm for Data Supply Chains (HCA-DSC), we compare HCA-DSC with a Dictionary-Based Compression for Long Time-Series Similarity[21].

6.2 Discovery accuracy

To verify the discovery accuracy of SADA-KP, an experiment is conducted in a data supply chain of nodes ranging from 10 to 80. Figure 2 shows the discovery accuracy values of TBSS-TSD, SSH-TSDEL, and SADA-KP.

From Fig. 2, we observe that the discovery accuracy of SADA-KP decreases with the growth of the number of nodes. It also is found that the discovery accuracies of TBSS-TSD and SSH-TSDEL are lower than SADA-KP (e.g., when the number of nodes is 60, the discovery accuracies of TBSS-TSD and SSH-TSDEL are 75.03% and 57.09%, respectively, lower than SADA-KP, which reaches 82.03%). Thus, our methods for SADA-KP

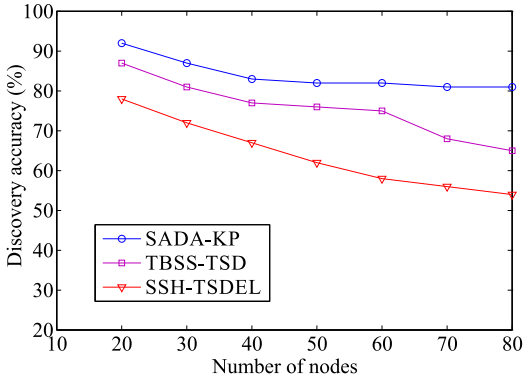


Fig. 2 Discovery accuracy comparison as a function of the number of nodes.

can provide more accurate discovery than traditional methods such as TBSS-TSD and SSH-TSDEL.

6.3 Query accuracy

To evaluate the query accuracy of the proposed approach, the total number of data supply chains, N, is set to 30 and 50, whereas the average length, M, of data supply chains ranges from 15 to 50. Figure 3 shows the query accuracy of HCA-DSC and DBC-TSS.

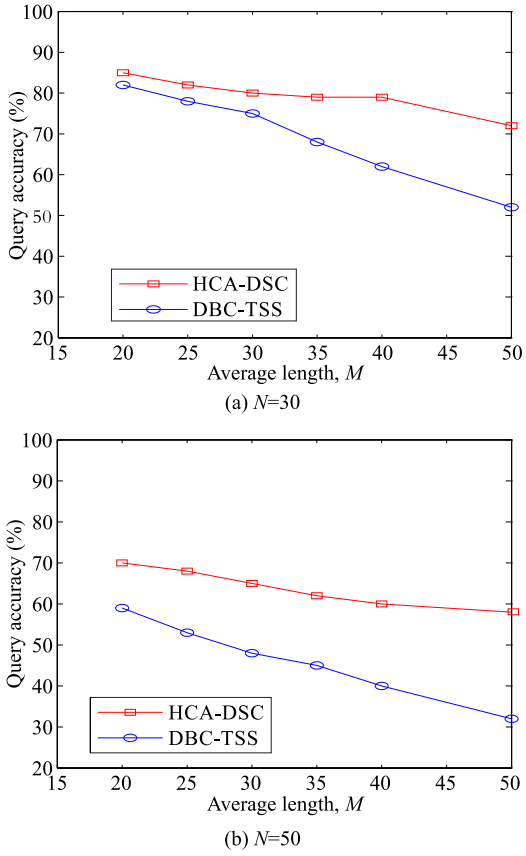


Fig. 3 Query accuracy comparison as a function of average length, *M*.

When the total number of data supply chains is set to 30, the main observation is that the query accuracy values range from 52% to 85.75%. Although DBC-TSS can present ideal results, its accuracy degrades rapidly with the increase of the average length of data supply chains and the lowest error rate is achieved at the maximum average length. The query accuracy of HCA-DSC performs better than DBC-TSS because it reduces the storage requirements, it potentially allows an efficient implementation of the similarity measurement and improves the quality of similarity search results. As shown in Fig. 3, there is a decreasing trend as the average length increases. We also observe that when the total number of data supply chains is set equal to 50, the corresponding query accuracy ranges from 32% to 68.3%. In general, when the value of M increases from 15 to 50, the query accuracy is higher when N is 30, than when N is 50. Data supply chains have an intrinsically high dimensionality, which necessitates the application of a dimensionality reduction technique. By using FSRM-KP, HCA-DSC can remove the effect of high dimensionality in the similarity search.

6.4 Query time

To evaluate the query time, we provide results for two algorithms, namely HCA-DSC and DBC-TSS. The total number of data supply chains is set to 30, 50, 70, and 90. Figure 4 shows the corresponding query times.

In Fig. 4, the first observation is that when the total number of data supply chains increases, the corresponding query time also increases. The reason for the increase is that, when searching a database for the most similar objects (data supply chains) to a given one, the abovementioned HCA-DSC and DBC-TSS algorithms have to compare a query object to

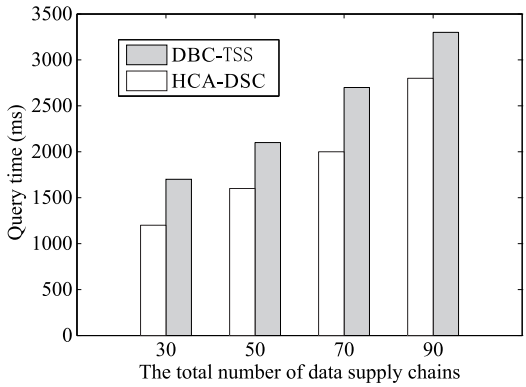


Fig. 4 Query time comparison, as a function of the number of data supply chains.

every object in a database. These approaches become prohibitive when the reference database is extremely large. Besides, the efficiency is affected by the number of objects in the database, since a distance measure is calculated for measuring the closeness of the corresponding objects. The second observation is that HCA-DSC performs consistently better than DBC-TSS. More specifically, when the number of data supply chains is equal to 90, HCA-DSC requires 2882 ms whereas the corresponding DBC-TSS requires 3350 ms. This is because HCA-DSC integrates the dimensionality reduction of data supply chains and subsequently the reduction of the search space. Hence, it improves the efficiency of the similarity search in the data supply chains datasets without sacrificing the quality of the results.

7 Conclusion

In this paper, we focus on a novel data supply chains similarity search problem. We first develop a feature space representation model based on key points, which can greatly reduce complex structures and the storage requirements. In addition, to measure the pairwise distances of the subsequences of data supply chains with high efficiency, we define a novel similarity measure based on multiscale features. Lastly, we propose a hierarchical clustering algorithm for data supply chains, which improves the quality of the similarity search results by identifying the most similar subsequences to a given query.

In our future work, we intend to establish a model of data supply chain performance evaluation based on the multidimensional evaluation index, which elucidates the performance of the data supply chain and reduces operating costs, further improving its competitive advantage

Acknowledgment

This work was partly supported by the National Natural Science Foundation of China (Nos. 61532012, 61370196, and 61672109).

References

[1] M. Ribeiro, K. Grolinger, and M. A. M. Capretz, MLaaS: Machine learning as a service, in *IEEE 14th International Conference on Machine Learning and Applications*, 2015, pp. 896–902.

[2] Z. H. Cheng, B. Yao, X. Wang, and Z. Zhou, Web service sub-chain recommendation leveraging graph searching, in 2014 *IEEE Computers, Communications and IT Applications Conference*, 2014, pp. 271–275.

[3] P. Groth, Transparency and reliability in the data supply chain, *IEEE Internet Computing*, vol. 17, no. 2, pp. 69–71, 2013.

[4] C. Ozturk, E. Hancer, and D. Karaboga, Dynamic clustering with improved binary artificial bee colony algorithm, *Applied Soft Computing Journal*, vol. 28, pp. 69–80, 2015.

[5] A. Hatamlou, Black hole: A new heuristic optimization approach for data clustering, *Information Sciences*, vol. 222, no. 3, pp. 175–184, 2013.

[6] B. Cui, Z. Zhao, and W. H. Tok, A framework for similarity search of time series cliques with natural relations, *IEEE Transactions on Knowledge & Data Engineering*, vol. 24, no. 3, pp. 385–398, 2012.

[7] H. Yin, S. Yang, M. A. Shaodong, F. Liu, and Z. Chen, A novel parallel scheme for fast similarity search in large time series, *China Communications*, vol. 12, no. 2, pp. 129–140, 2015.

[8] T. Iwashita, T. Hochin, and H. Nomiya, Optimal number of clusters for fast similarity search of time series considering transformations, in *2014 IIAI 3rd International Conference on Advanced Applied Informatics*, 2014, pp. 711–717.

[9] S. Ghassempour, F. Girosi, and A. Maeder, Clustering multivariate time series using hidden Markov models, *International Journal of Environmental Research & Public Health*, vol. 11, no. 3, pp. 2741–2763, 2014.

[10] T. Rakthanmanon and E. Keogh, Fast shapelets: A scalable algorithm for discovering time series shapelets, in *13th SIAM International Conference on Data Mining*, 2013, pp. 668–676.

[11] L. Karamitopoulos and G. Evangelidis, Cluster-based similarity search in time series, in *2009 4th Balkan Conference in Informatics*, 2009, pp. 113–118.

[12] P. Yue, L. Di, W. Yang, G. Yu, P. Zhao, and J. Gong, Semantic web services-based process planning for earth science applications, *International Journal of Geographical Information Science*, vol. 23, no. 9, pp. 1139–1163, 2009.

[13] S. Meng, W. Dou, X. Zhang, and J. Chen, KASR: A keyword- aware service recommendation method on mapreduce for big data applications, *IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 12, pp. 3221–3231, 2014.

[14] Z. B. Zhou, Z. Cheng, K. Ning, W. Li, and L. J. Zhang, A sub-chain ranking and recommendation mechanism for facilitating geospatial web service composition, *International Journal of Web Services Research*, vol. 11, no. 3, pp. 52–75, 2014.

[15] D. Singh and C. K. Reddy, A survey on platforms for big data analytics, *Journal of Big Data*, vol. 2, no. 1, pp. 1–20, 2015.

[16] B. Issac and W. J. Jap, Implementing spam detection using Bayesian and Porter Stemmer keyword stripping approaches, in *IEEE Region 10 Annual International Conference*, 2009, pp. 1–5.

[17] C. Stokes, A. Kumar, F. Choi, and R. Weischedel, Scaling

NLP algorithms to meet high demand, in *2015 IEEE International Conference on Big Data*, 2015, pp. 2839–2839.

[18] A. Andoni and K. Onak, Approximating edit distance in near-linear time, *ACM Symposium on Theory of Computing*, vol. 41, no. 6, pp. 199–204, 2011.

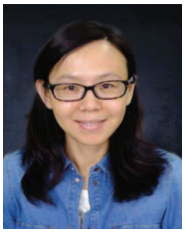
[19] M. Suntinger, H. Obweger, J. Schiefer, P. Limbeck, and G. Raidl, Trend-based similarity search in time-series data, in *2nd International Conference on Advances in Databases, Knowledge and Data Applications*, 2010, pp. 97–106.

[20] T. Nakamura, K. Taki, H. Nomiya, K. Seki, and K. Uehara, A shape-based similarity measure for time series data with ensemble learning, *Pattern Analysis and Applications*, vol. 16, no. 4, pp. 535–548, 2013.

[21] W. Lang, M. Morse, and J. M. Patel, Dictionary-based compression for long time-series similarity, *IEEE Transactions on Knowledge & Data Engineering*, vol. 22, no. 11, pp. 1609–1622, 2010.

Peng Li is currently a PhD candidate in Beijing University of Posts and Telecommunication. His main research interests include big data computation and communications. He received the master degree from Yanshan University, China, in 2014.

Hong Luo is a professor with the School of Computer Science, Beijing University of Posts and Telecommunications, China. She is also a research member of the Beijing Key Lab of Intelligent Telecommunication Software and Multimedia. Her research interests include Internet of Things, smart environments, data service, and communication software. She received the BS, MS, and PhD degrees from Beijing University of Posts and Telecommunications in 1990, 1993, and 2006, respectively.



Yan Sun is a professor with the School of Computer Science, Beijing University of Posts and Telecommunications, China. She is also a research member of the Beijing Key Lab of Intelligent Telecommunication Software and Multimedia. She obtained the BS degree from Beijing Jiaotong University in 1992, the MS and PhD degrees from Beijing University of Posts and Telecommunications in 1996 and 2007, respectively. Her research interests include Internet of Things, sensor networks, smart environments, and embedded systems.