Similarity-based Bayesian Learning from Semi-structured Log Files for Fault Diagnosis of Web Services

Xu Han\(^1\), Zhongzhi Shi\(^1\), Wenjia Niu\(^1\), Kunrong Chen\(^1\), Xinghua Yang\(^1\)
\(^1\)The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, 100190 Beijing, China
\(^2\)Graduate University of Chinese Academy of Sciences, 100049 Beijing, China
{hanxu, shizz, niuwenjia, chenkr, yangxh}@ics.ict.ac.cn

Abstract

With the rapid development of XML language which has good flexibility and interoperability, more and more log files of software running information are represented in XML format, especially for Web services. Fault diagnosis by analyzing semi-structured and XML like log files is becoming an important issue in this area. For most related learning methods, there is a basic assumption that training data should be in identical structure, which does not hold in many situations in practice. In order to learn from training data in different structures, we propose a similarity-based Bayesian learning approach for fault diagnosis in this paper. Our method is to first estimate similarity degrees of structural elements from different log files. Then the basic structure of combined Bayesian network (CBN) is constructed, and the similarity-based learning algorithm is used to compute probabilities in CBN. Finally, test data from input log files can be classified based on the generated CBN. Experimental results show our approach outperforms other learning approaches on those training datasets which have different structures.

1. Introduction

Today, increasing number of business functions are published as Web services by various organizations and companies. As a recommended standard, Business Process Execution Language (BPEL) \([1]\) is chosen for modeling business processes, whose execution can be supported by BPEL engines of different providers. Since there are more and more Web services are used in practice, high dependability and availability is becoming essential features for their execution. In order to enhance the dependability of service flow execution, self-healing of services is proposed in [2] which mainly includes fault diagnosis and recovery. As a critical step in self-healing process, the objective of fault diagnosis is to get possible fault reasons based on runtime information. Considering the fact that most detailed running information is recorded in log files, fault diagnosis by learning from log file data is becoming an important issue in this area.

As a semi-structured data format, XML is quickly becoming the standard means of information representation on the Internet. Because of its good flexibility and interoperability, more and more log files of software running information are represented in XML format, especially for Web services. Accordingly, fault diagnosis by analyzing semi-structured and XML like log files is considered as the key problem to be solved. Generally, semi-structured documents have much richer structural information than flat ones. Taking this into account, the main task of learning from this kind of documents will have more challenges than before. As to most learning methods for this kind of documents, there is a basic assumption that training data should be in identical structure. But this assumption does not hold in many situations in practice. Thus, a major challenge for semi-structured documents compared to flat ones is how to learn from the training data in different structures.

In this paper, we propose a similarity-based Bayesian learning approach for fault diagnosis. Our method is to first estimate similarity degrees of structural elements from different XML log files. Then the basic structure of combined Bayesian network (CBN) is constructed, and the similarity-based learning algorithm is used to compute probabilities in CBN. Finally, test data from input log files can be classified based on the generated CBN. Experimental results show our approach outperforms other learning approaches on those training datasets which have different structures.

The rest of this paper is organized as follows. Section 2 introduce the related work concerning fault diagnosis of Web services and Bayesian learning for
(semi-)structured documents. Section 3 provides an overview of the similarity-based Bayesian learning approach and the CBN model. The details of CBN generation, including how to compute probabilities of CBN using similarity-based learning algorithm, are presented in Section 4. Experimental results of this approach compared to other learning approaches are shown in Section 5. Finally, Section 6 draws the conclusion.

2. Related work

In recent years, some research work in Web services area has concentrated on how to enhance dependability of services. Self-healing services is a promising direction, which the International workshops QSWS and SHWS (2008 and 2009) were dedicated to. As a principal step of self-healing, fault diagnosis has attracted an increasing attention. From the methodology perspective, the existing work on fault diagnosis in this area can be mainly divided into two categories, including model-driven diagnosis and data-driven diagnosis.

With respect to the first category, the basic approach is to model the behavior and inner logic of the diagnosed service, and then discover runtime faults according to its model. The on-going work has been described in some published papers. WS-DIAMOND [3] is a European research project which eight research agencies have participated in. In this project, model-based diagnosis is adopted as the principal approach. Yan et al. [4, 5] present a model-based approach for diagnosing orchestrated Web service processes. In their approach, Web services with faults can be deduced from the variable dependency on execution trajectory, which is represented by the generated automata of BPEL description.

As to data-driven category, the diagnosis problem is usually transformed into the classification problem. Then it can be solved by using data mining and machine learning algorithms on log file data. As we know, many reported research efforts have focused on mining log files of computing systems. And there are some common places in log mining methods for regular computing systems and Web services. Because of this, for Web services the reference can be made to those existing methods. Considering the differences between two basic data types – plain text data and (semi-)structured data, mining approaches are often designed and implemented according to the type of training and test data. For plain-text log data, Li et al. use Naive Bayes [6, 7] to categorize text messages in log files, and utilize the temporal information to improve classification performance. In [8], Naive Bayes classifier, semi-supervised learning, and decision trees, are used to automatically recognize symptoms of recurrent faults. As for (semi-)structured log data, there are also some corresponding classification approaches. In [9], a database of failure signatures against which undiagnosed failure data can be matched, is constructed from monitoring data. And then anomaly-based clustering method is proposed to generate right clusters for diagnosing failures with low-confidence match. Denoyer and Gallinari [10] present an effective rule based classifier for XML data by using frequent discriminatory sub-structures within XML documents. As a Bayesian learning approach of semi-structured data, our approach is similar to that given by Denoyer and Gallinari in methodology aspect. But they are different in nature, because their generative model only concerns the training data in identical logic structure, whereas our approach aims at constructing the CBN model based on log data in different structures from heterogeneous sources (which will be discussed later in detail).

In addition, there are other diagnosis mechanisms and frameworks for Web services. In [12], a self-healing plug-in for BPEL engine is presented, which can enhance the ability of a standard engine to provide process-based recovery actions. It only provides the self-healing mechanism at infrastructure level without referring to diagnosis methods in detail. Ardissono et al. [13] propose a framework for Web Service orchestration which employs diagnostic services to support a fine grained identification of exception causes.

3. A Bayesian network approach to fault diagnosis

As a data-driven approach, the task of fault diagnosis is to construct the generative model based on the training data from semi-structured log files, and then classify test log data into possible fault categories using this model. On account of the heterogeneity of log file sources in practice, it is a new challenge to develop methods of learning from the semi-structured log data, which are similar in content but not identical in structure. For this reason, it is important to exploit the structural information contained in XML documents for learning task. According to the classical representation, an XML document can be considered as a tree in which each node represents a structural element. Given two XML log files generated by different service execution engines, the corresponding schema trees can be extracted from them, which often
have similar content but slightly different structures. In this paper, our main idea is to find the similarity between the nodes of these two trees, and then construct the generative model by learning from the training log data based on similarity degrees. Herein, we propose combined Bayesian network as a generative model for semi-structured log data, the probabilities of which depend on training data distribution of both corresponding elements and their counterparts with high similarity degrees.

### 3.1. Overview of the approach

In this approach, the learning problem is simplified by limiting the number of XML log files to two, which constitute the training dataset. (If the number of log files is more than two, the learning problem can be transformed into the equivalent one which consists of several two-file problems.) For example, suppose there are two log files $F_0$ and $F_1$, as shown in Figure 1. We can find that $F_0$ and $F_1$ have similar content but slightly different structures. In view of their structural differences, generating Bayesian network model from the training dataset is non-trivial. Herein, how to handle the similarity relation between schema elements of $F_0$ and $F_1$ can be taken as the key point. The objective of our approach is to generate the CBN by learning from all relevant data in training dataset based on quantified similarity.

![Figure 1. Fragments of two sample XML log files](image)

To give an overview of the fault diagnosis approach, we will firstly introduce its principal steps. Suppose the schema trees $T_0$ and $T_1$ are extracted from XML log files $F_0$ and $F_1$, then the main steps in this approach can expressed as follows:

1. Estimate the normalized similarity degrees of the nodes of $T_0$ and $T_1$, and then find the similar node pairs by ignoring those pairs whose degrees are under the threshold value;
2. Create the basic structure of combined Bayesian network, which includes combination part and private part;
3. For each schema tree, calculate the combination similarity degrees between its nodes and corresponding ones in CBN, and normalize the values of combination similarity degrees;
4. Compute the probabilities of the constructed CBN on all relevant data in training dataset using the similarity degrees obtained from steps (1) and (3);
5. Given a runtime log record in XML format, use the generative CBN model to classify the test log data, and diagnose possible faults according to the category it falls in.

As we see, steps (1), (3), and (4) play important role in this approach, which are responsible for the computation of similarity degrees. The detailed discussion of these steps will be given in section 4.

### 3.2. Modeling semi-structured log files with combined Bayesian networks

The Bayesian network is a suitable model for representing the dependencies and relations between different elements of semi-structured data. On account of structural differences of training data, we propose combined Bayesian network, which is modeled by learning from heterogeneous training data based on quantified similarity. As a generative model, CBN is capable of handling both structure and content information, and can be used to classify test log data.

Generally, each log file consists of a set of log data records, which will be labeled with related fault categories. These labeled log records constitute the training dataset. We associate a CBN model to each category of the training dataset. Since data records in same category may have different logical structures, a CBN is constructed by combining structural and content information of all data records in corresponding category. Then, the network parameters are learned from all training data records in this category. To realize fault diagnosis, we classify test log data records into possible fault categories by performing inference in constructed CBNs.

Consider the sample XML log files $F_0$ and $F_1$ in Figure 1. For the training data in a given category $c$, the schema information of each log record can be represented by $T_0$ or $T_1$. According to this, we can construct the basic structure of CBN by combining $T_0$ and $T_1$, which contains a set of structural nodes denoted by $Ns$. In addition, there are also a set of
content nodes (as the leaf nodes in CBN) denoted by \( M \) for representing textual information of log data. For simplicity, the discussion below will mainly focus on those structural nodes. Figure 2 shows the basic structure of CBN, which is a directed graph with \( T_0 \) and \( T_1 \) as its subgraphs. In fact, each node in this CBN has node \( c \) for the corresponding category as its father (which is omitted for sake of simplicity). According to quantified similarity results, the structural nodes can be divided into two groups, including combination group and private group. The combination group contains the nodes which have the corresponding nodes or their similar counterparts in both \( T_0 \) and \( T_1 \). In comparison with combination group, the private group is composed of the nodes, which only have the corresponding nodes in \( T_0 \) (or \( T_1 \)).

4. Similarity-based Bayesian learning for fault diagnosis

In this section, we propose a similarity-based Bayesian learning approach for fault diagnosis. Firstly, we introduce how to estimate similarity degrees of schema elements for constructing combined Bayesian networks. Then, we present how to learn probabilities of CBNs in detail. Finally, we give the method of classifying log data records using CBNs for fault diagnosis in subsection 4.3. To illustrate this learning approach, we will continue to use the example given in section 3.

4.1. Estimating similarity degrees for constructing CBNs

Since the probabilities of CBNs depend on the distribution of both corresponding data and their similar counterparts in training dataset, we need to find the correspondences between the nodes of extracted schema trees. Consider the schema trees \( T_0 \) and \( T_1 \) in previous example. There are a number of nodes in \( T_0 \) and \( T_1 \) which have similar meaning but different names. In order to discover the correspondences between these nodes, we will exploit both textual and structural information of \( T_0 \) and \( T_1 \).

The task of finding consistency between similar elements of two schemas is often regarded as matching. In this subsection, we will estimate element similarity degrees by schema matching. As for the matching method, we have made a reference to the similarity flooding algorithm [14], whereas our method utilizes graded structural information in different stages, which can improve both of initial mapping accuracy and computation efficiency. In the matching process, we take \( T_0 \) and \( T_1 \) as input, and produce similarity degrees between corresponding nodes of \( T_0 \) and \( T_1 \) as output. There are two steps in this process including: (1) initial matching, and (2) similarity propagation based on pairwise graph. In initial matching, we additionally exploit the structural information in attribute level. Given two nodes \( m_0 \) and \( n_1 \) of \( T_0 \) and \( T_1 \) which has attribute sets \( A_m \) and \( A_n \) respectively, we will match attributes of \( A_m \) and \( A_n \) as well as names of \( m_0 \) and \( n_1 \). The computation formula is

\[
Sim^0(m_0, n_1) = \theta \cdot \frac{|A_m \cap A_n|}{|A_m| + |A_n|} + (1 - \theta) \cdot \text{StrSim}(m_0, n_1)
\]

where \(|A_m|\) is the number of elements in set \( A_m \), \( \text{StrSim}(m_0, n_1) \) represents the similarity value of \( m_0 \) and \( n_1 \) by string matching, and \( \theta \) is a weight coefficient fixed by users. Based on initial matching results, we construct the pairwise graph in step (2). A portion of this pairwise graph is shown in Figure 3, where some nodes of low degrees in initial matching are ignored.

4.1.1. Then, similarity propagation is executed based on this graph. The computation formula for similarity propagation is

\[
Sim^{(i+1)}(n_0, n_1) = Sim^i(n_0, n_1) + \sum_{(m_0, m_1) \in \text{NeigSet}(n_0, n_1)} Sim^i(m_0, m_1) \cdot \omega((m_0, m_1), (n_0, n_1))
\]

where \( Sim^i(n_0, n_1) \) indicates the similarity degree of pairwise node \( (n_0, n_1) \) in the \( i \)-th iteration,
\( \omega((m_0, m_1), (n_0, n_1)) \) denotes the weight of edge between the given nodes (whose value equals to that of 1 divided by the outgoing edge number of the source node for this edge), and NeigSet\((n_0, n_1)\) represents the set of neighbors of node \((n_0, n_1)\) in the pairwise graph. This computation process is performed iteratively to estimate similarity degrees between nodes of \(T_0\) and \(T_1\). After this process, the computed similarity degrees will be normalized to the range of \([0, 1]\).

Table 1 shows a portion of the similarity estimation result. If we set the value of filtering threshold to 0.1, the pairs whose similarity degree is no bigger than 0.1 will be deleted from the CBN.

<table>
<thead>
<tr>
<th>Node in (T_0)</th>
<th>Node in (T_1)</th>
<th>Similarity (Sim)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault</td>
<td>Fault</td>
<td>0.90</td>
</tr>
<tr>
<td>DetailEntries</td>
<td>DetailEntries</td>
<td>0.68</td>
</tr>
<tr>
<td>DetailEntries</td>
<td>FaultEntries</td>
<td>0.44</td>
</tr>
<tr>
<td>FaultEvent</td>
<td>faultEvent</td>
<td>0.19</td>
</tr>
<tr>
<td>SourceActivity</td>
<td>activity</td>
<td>0.19</td>
</tr>
<tr>
<td>Name</td>
<td>name</td>
<td>0.17</td>
</tr>
<tr>
<td>Type</td>
<td>type</td>
<td>0.17</td>
</tr>
<tr>
<td>Value</td>
<td>value</td>
<td>0.17</td>
</tr>
<tr>
<td>FaultCode</td>
<td>FaultCode</td>
<td>0.16</td>
</tr>
<tr>
<td>Description</td>
<td>FaultString</td>
<td>0.10</td>
</tr>
<tr>
<td>Scope</td>
<td>encodingURI</td>
<td>0.10</td>
</tr>
</tbody>
</table>

As shown in Table 1, an example of the matching results could be the correspondence between node “DetailEntries” and node “FaultEntries”.

4.2. Similarity-based learning algorithm for computing CBN probabilities

In this subsection, we will introduce how to compute CBN probabilities using similarity-based leaning algorithm. According to computed similarity degrees, the conditional probabilities of CBNs will be obtained.

Based on the similarity result, the combination part of CBN can be created as shown in Figure 4. Then, the conditional probabilities related to the nodes of this part will be learned from training data. In fact, the similar counterparts of log records in training dataset also influence corresponding probability distribution of the CBN. Taking the influence of the similar counterparts into consideration, we will calculate the probabilities in the combination part of CBN, based on similarity degrees between nodes of \(T_1\) and CBN.

Before going into details of probability computation, it is necessary to introduce the concept of combination similarity degree \(CSim\), which represents the similarity between a node of \(T_1\) and its corresponding node (with the same name) of constructed CBN. The \(CSim\) is computed recursively from leaf to root nodes for each schema tree, using the following formula: (Herein, \(n\) and \(s\) are the CBN nodes which correspond to nodes \(n_i\) and \(s_i\) respectively.)

\[
CSim(n_i, n) = \frac{1 + \sum_{s \in SonsOf(n_i), s \in SonsOf(s)} CSim(s_j, s)}{1 + \max\{|SonsOf(n_i)|, |SonsOf(n)|\}}
\]

where \(SonsOf(n)\) denotes the set of son nodes for node \(n\) (and it is the same for node \(n_i\)). According to this, we can obtain the combination similarity degrees between nodes of \(T_0 (T_1)\) and CBN, as shown in Table 2.

Table 2. Combination similarity degrees between nodes of each schema tree and CBN

<table>
<thead>
<tr>
<th>Node in (T_0)</th>
<th>Node in CBN</th>
<th>Combination Similarity (CSim)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault</td>
<td>Fault</td>
<td>0.65</td>
</tr>
<tr>
<td>DetailEntries</td>
<td>DetailEntries</td>
<td>0.88</td>
</tr>
<tr>
<td>FaultCode</td>
<td>FaultCode</td>
<td>1</td>
</tr>
<tr>
<td>Description</td>
<td>Description</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

(a) \(CSim\) between nodes of \(T_0\) and CBN

<table>
<thead>
<tr>
<th>Node in (T_1)</th>
<th>Node in CBN</th>
<th>Combination Similarity (CSim)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fault</td>
<td>Fault</td>
<td>0.77</td>
</tr>
<tr>
<td>DetailEntries</td>
<td>DetailEntries</td>
<td>0.63</td>
</tr>
<tr>
<td>FaultEntries</td>
<td>FaultEntries</td>
<td>1</td>
</tr>
<tr>
<td>FaultCode</td>
<td>FaultCode</td>
<td>1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

(b) \(CSim\) between nodes of \(T_1\) and CBN

For the example mentioned above, suppose \(R_0\) and \(R_1\) are the record sets of training data, from which \(T_0\) and \(T_1\) can be extracted. Let \(R^c_0\) and \(R^c_1\) be the subsets of \(R_0\) and \(R_1\) with fault category \(c\). Then, let \(R^c\) be the union of these two subsets, i.e., \(R^c = R^c_0 \cup R^c_1\). Accordingly, given a log record \(r\) from record set \(R\), we use \(SD(r, m, n)\) to denote the similarity degree of
tree node (corresponds to element) $m_i$ of $r_i$ and node $n$ of CBN. The similarity degree SD will play an important role in learning process, which can be computed by the following formula:

$$SD(r_i, m_i, n) =
\begin{cases}
    CSim(m_i, n), & m_i \in CSet(n) \\
    Sim(m_i, l) \cdot CSim_{l,i}(l, n), & m_i \in SimSet(l) \land l \in CSet(n) \\
    0, & \text{other}
\end{cases}$$

where $CSet(n)$ represents the set of tree nodes corresponding to node $n$ (with the same name), and $SimSet(l)$ denotes the set of nodes similar to a tree node $l$. As an important step in learning, the times of a specific node and its parent appearing in the record set will be counted. Herein, we use $SimNum(r, p, q)$ to represent the similarity-based number of node $q$ having node $p$ as its parent for record $r$. We compute the value of $SimNum(r, p, q)$ by the formula shown below:

$$SimNum(r, p, q) =
\begin{cases}
    \sum_{c, pa(c) \in NodeSet(r)} SD(r, pa(e), p) \cdot SD(r, e, q), & q \in N_i \\
    \sum_{pa(c) \in NodeSet(r)} SD(r, pa(e), p) \cdot StrEqual(e, q), & q \in N_i
\end{cases}$$

where $pa(e)$ denotes the parent node of node $e$ in record $r$, $NodeSet(r)$ is the node (or element) set of record $r$, and $StrEqual(e, q)$ means whether $e$ equals to $q$ by string matching.

For the CBN of category $c$, we suppose each node has node $c$ as its father. Thus, the root element of each record in this category has the same parent node $c$. The conditional probability can be computed as follows:

$$P(n = q \mid pa(n) = p, c) = \frac{\sum_{e \in R^c} SimNum(r, p, q)}{\sum_{u \in N_i \cup N_j, r \in R} \sum_{p, q} SimNum(r, p, u)}$$

Based on above computation formulas, the similarity-based learning algorithm is shown below.

**Algorithm** Similarity-based learning algorithm

**Input:** A training record set $R' = R_0 \cup R^c_i$ with category $c$; A CBN node set $N = N_i \cup N_j$.

**Output:** $P(q \mid p, c)$

1. for each node $q \in N$ do
2.   for each record $r \in R'$ do
3.     Calculate $SimNum(r, pa(q), q)$ and add it to $numSum$;
4.     for each node $u$ where $pa(u) = pa(q) \land u \neq q$ do
5.       Calculate $SimNum(r, pa(u), u)$ and add it to $denSum$;
6.   end for
7. end for
8. $P(q \mid p, c) = numSum / (numSum + denSum)$;
9. end for

Given the training records in a category $c$ and the node set of constructed CBN, we can compute the conditional probability $P(q \mid p, c)$ for each node $q$ and its parent $p$ in the CBN, using this learning algorithm.

### 4.3. Fault diagnosis by classifying log data with CBNs

By labeling the training log records from heterogeneous sources with related categories, fault diagnosis can be viewed as a classification problem for log records obtained in runtime. As mentioned above, we can achieve the goal of fault diagnosis by classifying log data using the proposed generative model. According to this, the main approach of fault diagnosis is to first construct the combination part of CBNs by matching schema trees, then compute the probabilities of CBNs from training data based on estimated similarity degrees, and finally classify the newly obtained log data into possible fault categories using generated CBNs.

In classification task, the CBN model plays a key role which is proposed as a generative model based on Bayesian networks. The log records in training dataset with the same category will share the parameters of a CBN. That is to say, there is a set of such parameters for each fault category. The similarity-based probabilities of the CBNs can improve the accuracy of the classification task. Then, the log records in testing dataset can be classified into possible fault categories, by calculating the probability that each category will generate the log data record. Given a category $c$ and a test log record $r_{test}$, we can estimate the conditional probability by the formula

$$P(r_{test} \mid c) = \prod_{n_i \in N_i} P(n_i \mid pa(n_i), c) \prod_{n_j \in N_j} P(n_j \mid pa(n_j), c).$$

Given the set of predefined categories, our objective is to assign most probable category labels to unlabeled log records, based on the likelihood inference in corresponding CBNs. According to computed conditional probabilities, we will choose the category $c_{MAP}$ which has the maximum posteriori probability value to label the test log record, as shown below.

$$c_{MAP} = \arg \max_{c \in C} P(c)P(r_{test} \mid c)$$

$$= \arg \max_{c \in C} \prod_{n_i \in N_i} P(n_i \mid pa(n_i), c) \prod_{n_j \in N_j} P(n_j \mid pa(n_j), c)$$

### 5. Experiments

In this section, the experimental results on both real and synthetic log datasets are presented, which implements the similarity-based Bayesian learning
approach for diagnosing faults occurred in Web services execution. We compare the classification results of our approach with those of other classifiers for semi-structured documents. Experimental results show our approach outperforms other learning approaches, including Bayesian network model (BN) and Bayesian network model Fisher (BN Fisher) [10], on those training datasets which have different structures.

5.1. Datasets

The log data used in our experiments are collected from our Web services platform which is supported by ActiveBPEL engine [15]. The log data are generated by execution engine and recorded in database as persistent storage. The raw log data have different levels including debug, information, warning, error, and fatal levels. In pre-processing step, we have implemented a monitoring module to extract fault related log data, which ignores the low level information (e.g. the information in debug level) of raw log data. To simulate the data obtained from heterogeneous sources, we represent a portion of these log data using different XML structure. Then, we get the training dataset by labeling each record of these XML log data. There are 30 Web services running on this platform which is taken as the testbed of our experiment. Since faults in application and middleware levels are the common causes of failures in Web services execution, we inject 95 such faults into running services instances. In addition, we implement a synthetic data generation program to simulate the creation of log data, based on the symptom database [16] for IBM WebSphere Application Server. We have generated 1000 pieces of log records from 11601 pieces of XML log records in this database. And the training dataset is obtained by labeling each data record according to the value of “symptomtype” attribute.

Table 3 shows the detailed information of the datasets used in evaluation. The real and synthetic datasets are further divided into two subsets in different structure, respectively. For each of these two datasets, the random 70-30 data split is used for training and testing. We select 65 percent of data records in training dataset, which have the structure different from those of the other portion. Then, to show the advantage of our approach, we choose only 30 percent of records in testing dataset, whose structure is the same to that of the major portion of training dataset.

There are some advantages to evaluate our approach on both real and synthetic log datasets. On one hand, we use the real dataset to validate the approach in practical situations. On the other hand, the synthetic dataset help us to study the effects of different kinds of structural patterns.

5.2. Experimental results

For evaluating the performance of corresponding approaches, we define accuracy as the proportion of log records that are correctly assigned to a category. The average accuracy is used as the measure in this experiment, which is the mean accuracy over all categories of the real and synthetic datasets.

The similarity-based Bayesian learning approach is implemented in Java, and runs on a Windows machine with a dual (1.86 GHz) core Intel processor. To show the advantage of our approach in classification task, we compare it with other classifiers designed for semi-structured documents, including Bayesian network model (BN) and Bayesian network model Fisher (BN Fisher). The average accuracy of three classification approaches is shown in Figure 5.

![Figure 5. Average accuracy results](image)

Experimental results show that our approach outperforms BN and BN Fisher, when the percentage of training data records with the structure different from the majority of testing dataset is increasing.

6. Conclusion

In this paper, we focus on fault diagnosis by
analyzing semi-structured log data. By transforming fault diagnosis problem into classification problem, we can utilize the corresponding classification methods to diagnose faults. We propose a similarity-based Bayesian learning approach for constructing combined Bayesian networks, which are used as generative model to classify fault related log data. Different from other approaches, it can learn from training data without identical structure. To realize fault diagnosis, our approach consists of three main steps: (1) estimate similarity degrees of structural elements from different log files, (2) construct the basic structure of CBNs by computing its probabilities using similarity-based learning algorithm, and (3) classify test log data into possible fault categories based on the generated CBNs. Experimental results show that our approach outperforms BN and BN Fisher on those training datasets which have different structures.

This paper just presents our work as the first step in fault diagnosis of Web services by analyzing log data. In the future, we will enlarge the size of real training dataset and do more experiments on the extended dataset. Moreover, for further studies, we are to utilize some mechanisms to improve the efficiency of this approach while keeping the accuracy of diagnosis results.

Acknowledgement

This work is supported by National Natural Science Foundation of China (No. 60775035, 60903141, 60933004, 60970088), 863 National High-Tech Program (No. 2007AA10Z132), National Basic Research Priorities Programme (No. 2007CB311004), National Science and Technology Support Plan (No. 2006BAC08B06), and Dean Foundation of GUCAS (No. O85101JM03).

References