An Approach to Identifying False Traces in Process Event Logs

Hedong Yang, Lijie Wen, and Jianmin Wang

School of Software, Tsinghua University, Beijing 100084, China
{yanghd06,wenlj00}@mails.tsinghua.edu.cn, jimwang@tsinghua.edu.cn

Abstract. By means of deriving knowledge from event logs, the application of process mining algorithms can provide valuable insight into the actual execution of business processes and help identify opportunities for their improvement. The event logs may be collected by people manually or generated by a variety of software applications, including business process management systems. However logging may not always be done in a reliable manner, resulting in events being missed or interchanged. Consequently, the results of the application of process mining algorithms to such "polluted" logs may not be so reliable and it would be preferable if *false traces*, i.e. polluted traces which are not possibly valid as regards the process model to be discovered, could be identified first and removed before such algorithms are applied. In this paper an approach is proposed that assists with identifying false traces in event logs as well as the cause of their pollution. The approach is empirically validated.

Keywords: process mining, event log, business process management, noise identification.

1 Introduction

Process mining provides a bridge between data mining and traditional model-driven Business Process Management (BPM) [10,11]. A business process (e.g. a purchase order, an insurance claim, etc.) is a sequence or network of tasks performed by humans or by machines to purposefully achieve a specific business goal. BPM provides supports, by affording methods, techniques, and softwares etc., for (re)design, deployment (system configuration and process enactment), and analysis of operational business processes as well as concerned resources (humans, machines, data, etc.) [9]. Generally speaking, a process-aware information system (PAIS) plays a key role during the whole life cycle of BPM as depicted in Fig. 1. By means of deriving knowledge from event logs manually collected or auto-generated by a PAIS [5], process mining, which behaves like the traditional data mining as depicted by the red arrow in Fig. 1, is generally seen as a critical tool to improve operational business processes iteratively. The first of three main classical applications of process mining is *model discovery*, which objective is to extract process models, the most important data of BPM, from event logs [10,11]. An example is to mine a process model shown in Fig. 3 given

J. Pei et al. (Eds.): PAKDD 2013, Part II, LNAI 7819, pp. 533–545, 2013.

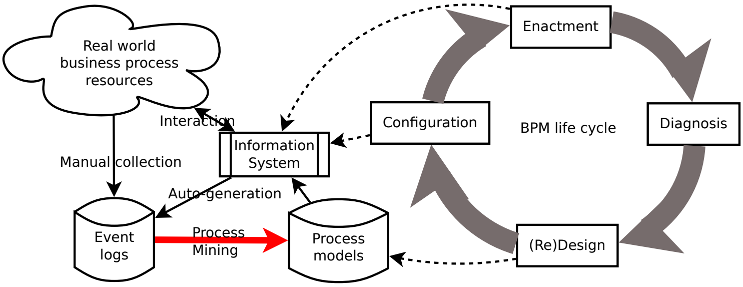

Fig. 1. Process mining and life cycle of Business Process Management

an event log containing four traces $\{ACDG, ADCG, BEH, BFH\}$. An overview of various mining algorithms for model discovery can be found in [9,14], and their implementations can be found in the open source platform, ProM[1], which has been used in industrial applications.

Generally speaking, the quality of process mining results is not determined only by the algorithm used but also by the quality of the concerned data, i.e. event logs which record the executions of process businesses. Of the criteria to judge the quality of an event log, one is *trustworthy* which requires recorded events and their orders being exactly same to what they happened [11]. However since event logs are often not treated as the key business data in real life, there are seldom policies to guarantee the quality of event logs. Consequently when a trace, a sequence of events recording an execution of a business process, was written into an event log, sometimes it was not recorded as it should be, namely it was *polluted*. For example, when deploying a PAIS in an organization for the first time, people have to describe the business processes of interest precisely and formally for configuring the system, which is often based on event logs collected manually where one or more events of a trace may be missed for some reasons. Another example happens daily in a hospital. Blood chemistry tests for patients are often carried out in groups rather than one by one instantly. And the test results are not available until some hours later, which are stamped with a date only. So do X-ray tests. Thus the sequence of such two tests may sometimes be messed up in the log for a patient since the granularity of timestamps is coarse-grained. Such traces that would not describe the actual executions of business processes, are called occurrences of *noise* [2] or *polluted traces*. The original sequence of events is transformed into another sequence of events by means of missing some events or interchanging the order of two events. In this paper we focus on these two types of *pollution*, which are widely accepted as the most common pollution of event logs (e.g., [3,6,15]). Although most problems in process mining have had satisfactory solutions, *noise identification* of logs is one of those unsolved which present impediments to advancement of the field [11].

Most of model discovery algorithms cannot guarantee the correctness of their mining results if the given event log is polluted. Mining algorithms can be classified

[1] `http://www.promtools.org`

into two categories. The first category consists of algorithms which assume the log to be noise-free (e.g., the most famous $\alpha-$algorithm [13]). For these algorithms it is necessary to identify occurrences of noise in the log and remove them before starting process mining. The second category consists of algorithms which have their own ways of dealing with occurrences of noise in the logs. These algorithms roughly treat low frequent traces as polluted ones directly or indirectly, no matter whether they are polluted or not, before process mining (e.g. [15]), during process mining (e.g. [1]) or after process mining (e.g. [2]). However, setting up a convincing threshold value is still a challenging problem.

Noise identification in process mining is similar to but not same as the data clean or outlier detection in data mining and the de-noising in signal processing. Traditional approaches for data clean make full use of relations among attributes and records of data [8], while there are unstructured traces only in event logs. Although a polluted trace is not the same as the normal trace which it should be, it may be by chance the same as another trace that is normal. Hence traditional approaches for outlier detection cannot be used [16]. Approaches for de-noising in signal processing focus on the Gaussian white noise, the widely accepted pollution type in the field, and are hard to be applied to deal with the pollution in process mining (e.g.,[4,7]). To summarize, algorithms available in these fields cannot be applied to identify polluted traces in an event log directly because of the characteristics of event logs and pollution concerned.

This then leads to the demand for a separate approach for noise identification. As a polluted trace may appear as another normal trace, we focus only on *false traces* in the paper, i.e. polluted traces which are not possibly valid as regards the process model to be discovered. Given a polluted log, as we do not have access to that process model that generated the log, we propose an approach, FATILP (FAlse Trace Identification based on Latent Probability), to helping find out false traces in a probabilistic manner, based on the occurrence frequencies of the observed traces and their transformation relations presented as a conditional probability matrix. The matrix describes the possible pollution type of the log, which itself can be obtained interactively by applying the approach.

It is important to note that our method for identifying false traces in a polluted log is not dependent on the choice of a specific mining algorithm. Our results can directly be used for those algorithms that are sensitive to false traces in logs (e.g. [6,13]). Beyond the field of process mining, the approach may be applied in the field of data provenance (e.g. to find out the origin of data), social network (e.g. to estimate the evolution of a social network), or traditional data mining (e.g. to mine the occurrence patterns of hot topics on the web).

The remainder of this paper is organized as follows. Section 2 describes basic concepts needed to define the problem and to describe our approach, explains three reasonable assumptions needed by our approach and formulates the problem of false trace identification of event logs for process mining. Then the proposed approach for the false trace identification problem is outlined in Section 3. In Section 4 the results obtained are evaluated and examined in an experimental manner. Section 5 concludes the paper and outlines future work.

2 Problem Characterization

2.1 Basic Definitions

A *task* is an activity to be performed in the context of a business process. A *process model* provides an abstraction of a business process capturing its tasks and all possible execution orders of these tasks in a formal manner. A *process instance* represents an actual execution of a business process. A *trace* is the result of the successful completion of a process instance and consists of a sequence of events, where each *event* corresponds to the execution of a task and all events are totally ordered typically on the basis of the timestamps that they were recorded. An *event log* is a set of traces, which records executions of a process model [12].

Two traces are *equivalent* if and only if their lengths are equal and every event of the first trace refers to the same task as the corresponding event at the same position of the second trace. A *trace class* consists of traces equivalent to each other. For simplicity, we refer to *a trace as a sequence of task names* to which the events of the trace correspond respectively, and thus a log as a *bag* of traces generated by a process model. As mentioned before, a trace is referred to as a *polluted trace* if it does not describe the actual execution of a business process, and as a *normal trace* otherwise. As a polluted trace may appear as a normal one, we define a special kind of polluted trace as follows.

Definition 1 (False Trace). *Given a process model P and a log L. A trace σ of L is referred to as a* false trace *if and only if it is not equivalent to any normal trace of P.*

A trace is referred to as a *true trace* if it is not a false trace. All normal traces are true traces and all false traces are polluted traces. Some polluted traces may be same as true traces. As illustrated in Fig. 2, a normal trace T_6 may be transformed into some polluted traces, and an observed trace T_2 in a log may originate from some normal traces. Traces T_0, T_1, T_8 and T_9 are false traces. Obviously the concepts of event log and false trace are quite different from those of *trajectory data* and *outlier* in the field of data mining respectively.

2.2 Assumptions

In this subsection the assumptions, which precisely characterise the event log and pollution type on the one hand and underpin the proposed approach on the other hand, are made explicit. Each assumption is described in detail and it is argued that the assumption is reasonable, why it is needed, and what would go wrong if the assumption was not made.

Assumption 1. *Normal traces occur randomly and independently.*

By observing the execution log, it is not possible to determine what the next trace will be recorded, based on the observed traces. It is reasonable to assume that traces appear randomly and independently.

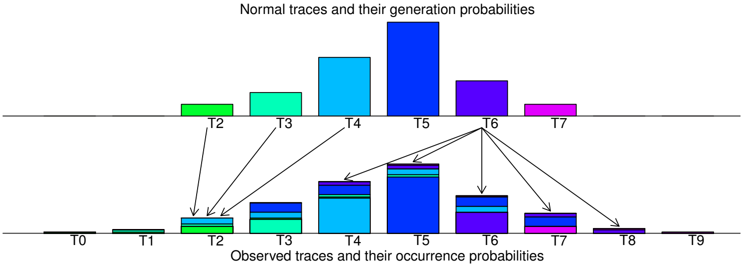

Fig. 2. Generation probabilities and occurrence probabilities of traces

If the occurrence of a new trace depends on an observed trace, the new trace and the observed trace are correlated. We treat them as different occurrences of the same trace as they can be (partially) deduced from existing ones.

Assumption 2. *A normal trace occurs with a constant but unknown probability, which may vary across different traces.*

A trace represents a particular application scenario of a process model. When a business process has been running for years, the same scenario may appear periodically. As time goes by the *occurrence frequencies* of the traces become relatively stable and in the long run they may converge to constant values, i.e. to their *latent generation probabilities.* Note that because of pollution, the generation probability of a trace is generally different from its occurrence probability.

If this assumption does not hold, we cannot solve the problem of noise identification of an event log without further information about the occurrence of traces. It is worthwhile noting that this means that our approach does not work so well for logs that result from processes that have not been running for a very long time as trace occurrence frequencies may not have sufficiently stabilized.

Assumption 3. *The pollution occurs randomly and independently, and given a normal trace the conditional probability of transforming the normal trace to another polluted trace because of pollution is a constant value, which may vary across different polluted traces.*

According to our observations, it is general that all traces in a log are not polluted, and that the pollution of a normal trace seldom depends on previous occurrences of pollution. Thus it is reasonable to assume the random and independent occurrence of pollution. Note all possible polluted traces of a normal trace are determinate because of its determinate conditional probabilities.

The assumption reflects the key idea of the proposed approach, i.e. trying to mimic the process of pollution and then to identify false traces by making full use of the relationships between false traces and their corresponding normal traces, which can be presented as a conditional probability matrix, i.e. so-called a *pollution matrix.* Such relationship may be various, yet we here require its conditional transformation probability to be constant. Without detail information of pollution, it is typically assumed that all possible polluted traces of a normal trace have the same conditional transformation probability. A priori knowledge of pollution may help set up the probability value for a specific transformation.

2.3 Problem Formulation

In this paper we are concerned with finding answers to the following problems related to a polluted event log.

Problem 1 (False trace identification problem). Given a polluted log L of an unknown process model, and a pollution matrix $\mathbf{M}$, which traces among all traces in L are most likely to be false traces?

Problem 2 (False trace discovery problem). Given a polluted log L of an unknown process model. Among all traces in L which are most likely to be false traces?

3 Approach

3.1 Key Idea

Given an event log L, for all observed traces $T_1, T_2, \cdots, T_M$ their *occurrence frequencies* $\mathbf{F} = \{f_1, f_2, \cdots, f_M\}$ are defined by $f_i = n_i/N$ for all $1 \leq i \leq M$, where $N = \sum_{i=1}^{M} n_i$ and n_i is the occurrence time of T_i. Based on all assumptions presented in subsection 2.2, suppose there are all W possible traces. Let $\mathbf{G} = \{g_1, g_2, \cdots, g_M\}$ be the latent generation probabilities of the observed traces, $p_{i,j} = \mathit{Prob}(T_j|T_i)$ the conditional probability of transforming trace T_i to T_j where $1 \leq i \leq M$ and $1 \leq j \leq W$, and $\mathbf{P} = \{p_1, p_2, \cdots, p_W\}$ the occurrence probability of the traces either observed or unobserved. Especially the combined conditional transformation probability of all unobserved traces of T_i is $u_i = \sum_{j=M+1}^{W} p_{i,j} = 1 - \sum_{j=1}^{M} p_{i,j}$. All these conventions are presented in Table 1, and the probabilities are in gray since they are unknown.

Formally, the relationship between $\mathbf{G}$ and $\mathbf{P}$ can be described as

$$p_j = \sum_{i=1}^{M} g_i * p_{i,j}, \text{ where } 1 \leq j \leq W \ . \tag{1}$$

The start point of the proposed approach, FATILP (FAlse Trace Identification based on Latent Probability), is the occurrence frequencies of traces $\mathbf{F}$, which will converge to the occurrence probabilities of the traces $\mathbf{P}$ respectively according to the law of large numbers in probability theory. If $\mathbf{P}$ can be estimated based on $\mathbf{F}$, the $\mathbf{G}$ can be calculated by means of the equation (1). As we know a process model does not generate a false trace, which implies the latent generation probability of a false trace should be *zero*. Thus the false traces can be identified based on their latent generation probabilities. Precisely, the FATILP consists of three steps as follows.

1. Process the given log to derive occurrence frequencies of observed traces.
2. Estimate the latent generation probabilities of observed traces by means of minimizing a distance function. This is the key step of the approach.
3. Perform a χ^2 test on the estimation result, and identify false traces.

Table 1. Pollution matrix and event log information

Trace	T_1	T_2	$\cdots$	T_M	T_{M+1}	$\cdots$	T_W	prob.unobserv	Gen. prob.
T_1	$p_{1,1}$	$p_{1,2}$	$\cdots$	$p_{1,M}$	$p_{1,M+1}$	$\cdots$	$p_{1,W}$	u_1	g_1
T_2	$p_{2,1}$	$p_{2,2}$	$\cdots$	$p_{2,M}$	$p_{2,M+1}$	$\cdots$	$p_{2,W}$	u_2	g_2
$\vdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\cdots$	$\vdots$	$\vdots$	$\vdots$
T_M	$p_{M,1}$	$p_{M,2}$	$\cdots$	$p_{M,M}$	$p_{M,M+1}$	$\cdots$	$p_{M,W}$	u_M	g_M
Occur. prob.	p_1	p_2	$\cdots$	p_M	p_{M+1}	$\cdots$	p_W	p_U	
Occur. freq.	f_1	f_2	$\cdots$	f_M	f_{M+1}	$\cdots$	f_W	0	
Occur. times	n_1	n_2	$\cdots$	n_M	n_{M+1}	$\cdots$	n_W	0	

3.2 False Traces Identification

Since there is often an error between the **P** and **F**, it is not appropriate to replace **P** with **F** directly. Here we define as follows a distance function Q^2 between **P** with **F** . The minimization of the distance function would force **P** of false traces to be zero or be very near to zero since the unobserved traces have higher weights, and consequently **G** of false traces would be zero or very near to zero since **P** and transform probability are non-negative (refer to equation (1) for detail).

$$\underset{\mathbf{G}}{\operatorname{argmin}}\, Q^2 = \underset{\mathbf{G}}{\operatorname{argmin}}\, p_U^2 N^2 + \sum_{i=1}^{M} (f_i - p_i)^2 \frac{N}{f_i} = \underset{\mathbf{G}}{\operatorname{argmin}} (\sum_{j=1}^{M} g_j * u_j)^2 N^2 + \sum_{i=1}^{M} (f_i - \sum_{j=1}^{M} g_j * p_{j,i})^2 \frac{N}{f_i} \qquad (2)$$

subject to $\sum_{i=1}^{M} g_i = 1$ and $g_i \geq 0$.

Now the false traces identification problem has been modeled as a quadric optimization problem, whose computation complexity is determined by the number of variables and the number of constraints. From equation (2), it is known that there are M variables and $M + 1$ constraints. To best of our knowledge, 32,000 and 16,000 are the limits of numbers of variables and constraints for a non-linear optimization problem respectively, achieved by the Lingo System (version 12.0).[2] Those are enough for almost all false trace identification problems we believe.

Once **G** are obtained, observed traces with latent generation probabilities lower than one tenth of the smallest occurrence frequency among all observed traces, an objective threshold we proposed, may be false traces. The acceptance of the identification result depends on the result of a χ^2 test with a specified confidence level $1 - \alpha$. If the test fails, which indicates that **F** cannot reflect **P** of traces, the identification result should be rejected. It is necessary to note that passing χ^2 test is not a sufficient condition but a necessary condition.

3.3 False Trace Discovery

It is general that only partial information about pollution is known. For example, an observed trace is found being polluted by chance, but it is unknown what

[2] `http://www.lindo.com/`

the ratio of polluted traces versus observed traces is. Yet the FATILP can still help discover possible false traces by trying all types of known pollution and valid pollution ratios with the partial information. Two heuristic rules are used to find a better pollution description: 1) The smaller the distance, the better the pollution description. 2)The smaller the distance, the better the pollution ratio.

We believe that the false traces discovery is an interactive process. At first the FATILP is run with transformation matrix, elements of which are initialized either based on partial pollution information or with equal transformation probability. After the estimated results have been analyzed, the information about the pollution improved, and the element values of the transformation matrix revised, the FATILP will be run again. Iteratively the most possible pollution type of the log, the appropriate pollution ratio and all possible false traces will be found out at last. The FATILP is an indispensable tool during the interaction.

4 Experiments

Experiments were carried out to 1) validate the proposed approach, 2) demonstrate how to discover an appropriate pollution ratio as well as 3) the most possible pollution type for a given polluted log.

4.1 Experiment Design

An experiment consists of two steps, 1) generating logs according to the specified generation probabilities of normal traces, pollution type, pollution ratio and log length, and 2) identifying false trace as well as evaluating experiment results.

For a process model shown in Fig. 3, there are four normal traces $T_1(ACDG)$, $T_2(ADCG)$, $T_3(BEH)$, and $T_4(BFH)$. For these traces, we here define three typical generation probability distributions of normal traces as shown in Table 2.

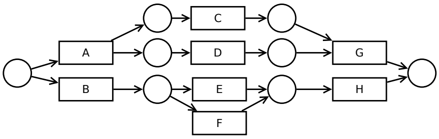

Fig. 3. A simplified business process model

Table 2. Generation probability distributions

Type	$P(T_1)$	$P(T_2)$	$P(T_3)$	$P(T_4)$
B(balanced)	0.25	0.25	0.25	0.25
N(unbalanced)	0.59	0.35	0.05	0.01
I(ext-unbalanced)	0.6999	0.25	0.05	0.0001

Two types of pollution are simulated, pollution D that $R\%$ of traces are polluted by missing an event and every event of a trace may be missed with the same probability and pollution E that $R\%$ of traces are polluted by exchanging orders of two adjacent events and every pair of adjacent events of a trace may be exchanged with the same probability. It is necessary to note that although complicated pollution, e.g. the two elementary types being combined together and/or repeated some times, may lead to diverse element values of the transformation matrix, this diversity can be approximated by means of elementary pollution along with various generation probability distributions of traces (refer to the Q^2). That is the reason why pollution types D and E are selected.

To evaluate the performance of approaches for the false trace identification problem, the correct identification rate, $h = \frac{tp+tn}{tp+fn+tn+fp}$, is defined, where tp and fn are the numbers of false traces being identified as false traces and as true traces respectively, tn and fp are the numbers of true traces being identified as true traces and as false traces respectively. Each experiment is repeated 100 times on 100 logs and the average values are used for evaluation.

To distinguish one experiment from the others, we name an experiment with a code $XYZK$, where X is a pollution type (D or E), Y is a pollution ratio ($Y \times 10\%$), Z is a generation distribution type (B,N or I), and K is sample size, i.e. log length. The K may be omitted when the sample size is $5k$.

4.2 Experiment Results

In Fig. 4, both sub-figures (a) and (b) depict that the performance of the proposed approach decreases from on balanced logs to on extreme unbalanced logs, but values of best performance of all experiments are greater than 0.9. The approach is so sensitive to the pollution ratio that the best identification rates can only be achieved around the real pollution ratio, 50%, no matter what the pollution type is and what the distribution is. Both sub-figures (c) and (d) show that the performance of the approach gets better when the length of the extreme unbalanced log increases. We can conclude that if the log length is big enough, the performance of the approach on the extreme unbalanced logs would be as good as that on balanced logs, and there is no significant difference between performance of the proposed approach on logs with E and that on logs with D. Thus to illustrate the performance of the approach, experiments on logs with any type of probability distributions and any type of pollution are acceptable.

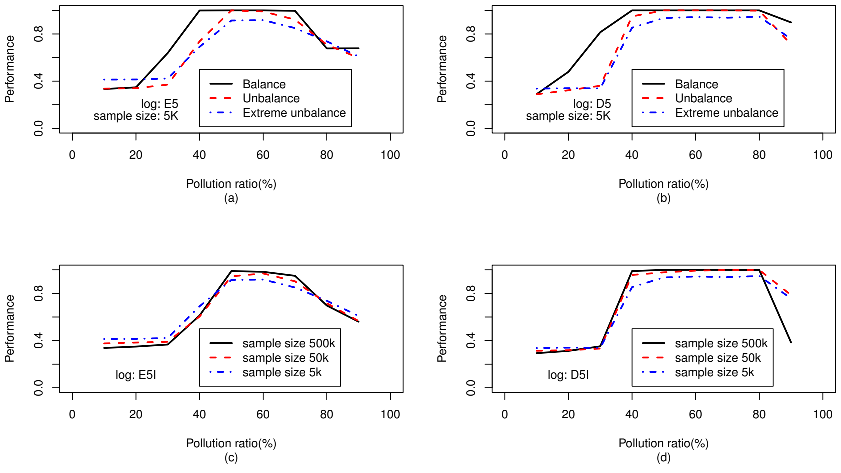

Fig. 4. Approach performance

Table 3. Performance comparison

	D5B	D5N	D5I	D5I50k	D5I500k	E5B	E5N	E5I	E5I50k	E5I500k
FATILP	1.00	1.00	0.94	0.98	1.00	1.00	1.00	0.91	0.95	0.99
Thr	1.00	0.87	0.88	0.86	0.87	1.00	0.83	0.87	0.83	0.83

The traditional approach for noise identification in process mining is denoted as "Thr" in Table 3, which depends on an empirical threshold [15]. The results of "Thr" are based on the most appropriate threshold values respectively. Although the FATILP works as well as "Thr" on balanced logs, it works better than "Thr" on unbalanced logs. It is interesting that the performance of FATILP increases when the log length increases, while the "Thr" does not. The main reason, we believe, is that the proposed approach considers the nature of pollution by means of modeling the process of pollution of event logs in a probabilistic manner.

Table 4. Average Q^2/h of each experiment on E polluted balanced logs

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
$R_A = .1$	0.00/1.00	0.01/1.00	0.01/0.67	0.02/0.37	0.02/0.33	0.03/0.33	0.03/0.33	0.04/0.33	0.05/0.16
$R_A = .2$	0.10/1.00	0.00/1.00	0.01/1.00	0.03/0.64	0.04/0.33	0.06/0.33	0.07/0.33	0.08/0.33	0.10/0.16
$R_A = .3$	0.37/1.00	0.05/1.00	0.00/1.00	0.02/0.98	0.05/0.63	0.08/0.34	0.11/0.33	0.14/0.23	0.18/0.16
$R_A = .4$	0.84/1.00	0.21/1.00	0.04/1.00	0.00/1.00	0.03/1.00	0.08/0.65	0.14/0.37	0.19/0.18	0.29/0.16
$R_A = .5$	1.44/1.00	0.46/1.00	0.15/1.00	0.03/1.00	0.00/1.00	0.03/1.00	0.13/0.91	0.25/0.49	0.41/0.16
$R_A = .6$	1.65/0.50	0.70/0.50	0.34/0.99	0.13/1.00	0.03/1.00	0.00/1.00	0.03/1.00	0.16/1.00	0.43/0.66
$R_A = .7$	1.38/0.18	0.67/0.50	0.43/0.62	0.26/0.67	0.12/0.99	0.03/1.00	0.00/1.00	0.04/1.00	0.23/1.00
$R_A = .8$	1.00/0.00	0.65/0.50	0.48/0.66	0.34/0.66	0.22/0.67	0.11/0.88	0.03/1.00	0.00/1.00	0.06/1.00
$R_A = .9$	0.82/0.12	0.68/0.48	0.54/0.66	0.41/0.66	0.30/0.68	0.20/0.73	0.11/0.83	0.04/1.00	0.00/1.00

Table 5. Discovering pollution type on D polluted logs

	h_E	Q^2_E	χ^2_E	h_D	Q^2_D	χ^2_D
$R_A = 0.1$	0.27	0.09	$4.7E + 2$	0.29	0.05	$2.4E + 2$
$R_A = 0.2$	0.27	0.20	$1.0E + 3$	0.47	0.09	$4.4E + 2$
$R_A = 0.3$	0.27	0.33	$1.8E + 3$	0.82	0.10	$5.1E + 2$
$R_A = 0.4$	0.28	0.46	$2.7E + 3$	1.00	0.04	$2.2E + 2$
$R_A = 0.5$	0.28	0.62	$4.1E + 3$	1.00	0.00	$1.1E + 1$
$R_A = 0.6$	0.32	0.78	$6.4E + 3$	1.00	0.04	$2.2E + 2$
$R_A = 0.7$	0.34	0.95	$1.1E + 4$	1.00	0.16	$9.7E + 2$
$R_A = 0.8$	0.54	1.10	$2.9E + 4$	1.00	0.37	$2.9E + 3$
$R_A = 0.9$	0.60	1.19	$2.E + 22$	0.89	0.63	$1.0E + 4$

Table 4 contains average least distances (Q^2 values) and average performance (h values) of experiments on E polluted balanced logs. The upper line lists pollution ratios used to generate polluted logs. The left column lists assumed pollution ratios used to identify false traces. From the values in the table, we know that both the minimal value of distance is obtained and the best performance is achieved when the assumed pollution ratio equals the real ratio. This property can help discover the correct pollution ratio among assumed ratios, with which the approach performs best on a log given correct pollution type.

An example of looking for the exact pollution type as well as pollution ratio of a polluted log is presented in Table 5. A balanced log is polluted by means of pollution D with ratio 50%. To find out the real pollution, first the pollution E is assumed with pollution ratio increasing from 10% to 90%. The pollution ratio 10% seems a good choice since values of the distance Q_E^2 and the statistic χ_E^2 are minimal respectively. Second the pollution D is tried with, where both Q_D^2 and χ_D^2 reach their minimal values at ratio 50%. And both Q_D^2 and χ_D^2 with ratio 50% are less than Q_E^2 and χ_E^2 with ratio 10%, and the D pollution with ratio 50% may be a good option. Furthermore, the value of χ_D^2 with ratio 50% is 11, which is much smaller than the critical value $43.82 (= \chi^2(19))$ with a confidence level 99.9%. Therefore the pollution D with ratio 50% is acceptable. Thus the proposed approach help find out the real pollution type of a polluted log.

5 Conclusions and Future work

In this paper, the noise identification problem of event logs for process mining was discussed. We distinguished the concept of false trace, i.e. the invalid traces as regards the process model to be discovered, from that of the polluted trace, i.e. noise, and focused on the false trace identification problem. On some natural and reasonable assumptions, we characterised the problem and modeled it as a quadric optimization problem of estimating a probability distribution. Then we proposed a common approach, FATILP, to estimate the latent generation probability distribution of normal traces given a polluted log and a description of pollution, and then to identify false traces at a user-specified confidence level. Experiment results show that the proposed approach works better than traditional approaches, and it can be applied not only to identify false traces in a polluted logs but also to discover the most possible pollution type of the log as well as an appropriate pollution ratio interactively.

The work presented in this paper may be extended in several directions. First, the approach may be improved by taking informative completeness of event logs into consideration. Second, it may be possible to improve the precision of the estimation of latent generation probabilities of observed traces. Third, the approach may be extended to deal with new types of pollution, e.g. duplicate records of an event.

Acknowledgements. The authors are indebted to Professor Wil M. P. van der Aalst for his suggestions on the motivation of the topic, to Professor Arthur H. M. ter Hofstede for his advice on the topic, constructive suggestions, and continuous encouragement, and to Dr. Moe T. Wynn for her suggestions on the presentation of the paper.

The work reported in this paper is partially supported by the 973 Plan Program of China (No. 2009CB320700), the 863 Plan Program of China (No. 2012AA040911 and No. 2012AA040904) and NSF Projects of China (No. 61073005 and No. 61003099).

References

1. Aires da Silva, G., Ferreira, D.R.: Applying hidden markov models to process mining. In: Rocha, A., Restivo, F., Reis, L.P., Ao, S.T. (eds.) Sistemas e Tecnologias de Informação: Actas da 4a. Conferência Ibérica de Sistemas e Tecnologias de Informação, pp. 207–210. AISTI/FEUP/UPF (2009)
2. Alves de Medeiros, A.K., Weijters, A.J.M.M., van der Aalst, W.M.P.: Genetic process mining: an experimental evaluation. Data Mining and Knowledge Discovery 14(2), 245–304 (2007)
3. Cook, J., Du, Z., Liu, C., Wolf, A.: Discovering models of behavior for concurrent workflows. Computers in Industry 53(3), 297–319 (2004)
4. Donoho, D.: De-noising by soft-thresholding. IEEE Transactions on Information Theory 41(3), 613–627 (1995)
5. Dumas, M., van der Aalst, W.M.P., ter Hofstede, A.H.M. (eds.): Process-Aware Information Systems: Bridging People and Software through Process Technology. Wiley Interscience, Hoboken (2005)
6. Maruster, L., Weijters, A.J.M.M., van der Aalst, W.M.P., Bosch, A.: A Rule-Based Approach for Process Discovery: Dealing with Noise and Imbalance in Process Logs. Data Min. Knowl. Discov. 13(1), 67–87 (2006)
7. Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P.: Image denoising using scale mixtures of gaussians in the wavelet domain. IEEE Transactions on Image Processing 12(11), 1338–1351 (2003)
8. Rahm, E., Do, H.H.: Data cleaning: Problems and current approaches. IEEE Data Eng. Bull. 23(4), 3–13 (2000)
9. van der Aalst, W.M.P.: Process Mining: Discovery, Conformance and Enhancement of Business Processes. Springer, Berlin (2011)
10. van der Aalst, W.M.P.: Process mining: Overview and opportunities. ACM Trans. Management Inf. Syst. 3(2), 1–17 (2012)
11. van der Aalst, W., Adriansyah, A., de Medeiros, A.K.A., Arcieri, F., Baier, T., Blickle, T., Bose, J.C., van den Brand, P., Brandtjen, R., Buijs, J., Burattin, A., Carmona, J., Castellanos, M., Claes, J., Cook, J., Costantini, N., Curbera, F., Damiani, E., de Leoni, M., Delias, P., van Dongen, B.F., Dumas, M., Dustdar, S., Fahland, D., Ferreira, D.R., Gaaloul, W., van Geffen, F., Goel, S., Günther, C., Guzzo, A., Harmon, P., ter Hofstede, A., Hoogland, J., Ingvaldsen, J.E., Kato, K., Kuhn, R., Kumar, A., La Rosa, M., Maggi, F., Malerba, D., Mans, R.S., Manuel, A., McCreesh, M., Mello, P., Mendling, J., Montali, M., Motahari-Nezhad, H.R., zur Muehlen, M., Munoz-Gama, J., Pontieri, L., Ribeiro, J., Rozinat, A., Seguel Pérez, H., Seguel Pérez, R., Sepúlveda, M., Sinur, J., Soffer, P., Song, M., Sperduti, A., Stilo, G., Stoel, C., Swenson, K., Talamo, M., Tan, W., Turner, C., Vanthienen, J., Varvaressos, G., Verbeek, E., Verdonk, M., Vigo, R., Wang, J., Weber, B., Weidlich, M., Weijters, T., Wen, L., Westergaard, M., Wynn, M.: Process mining manifesto. In: Daniel, F., Barkaoui, K., Dustdar, S. (eds.) BPM Workshops 2011, Part I. LNBIP, vol. 99, pp. 169–194. Springer, Heidelberg (2012)
12. van der Aalst, W.M.P., van Hee, K.M.: Workflow Management: Models, Methods, and Systems. MIT Press, Cambridge (2004)
13. van der Aalst, W.M.P., Weijters, A.J.M.M., Maruster, L.: Workflow Mining: Discovering Process Models from Event Logs. IEEE Transactions on Knowledge and Data Engineering 16(9), 1128–1142 (2004)

14. van Dongen, B.F., Alves de Medeiros, A.K., Wen, L.: Process Mining: Overview and Outlook of Petri Net Discovery Algorithms. In: Jensen, K., van der Aalst, W.M.P. (eds.) ToPNoC II. LNCS, vol. 5460, pp. 225–242. Springer, Heidelberg (2009)
15. Weijters, A.J.M.M., van der Aalst, W.M.P.: Rediscovering Workflow Models from Event-Based Data Using Little Thumb. Integrated Computer-Aided Engineering 10(2), 151–162 (2003)
16. Xiong, H., Pandey, G., Steinbach, M., Kumar, V.: Enhancing Data Analysis with Noise Removal. IEEE Transactions on Knowledge and Data Engineering 18(3), 304–319 (2006)