Expert Systems with Applications 40 (2013) 3722–3737

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

A comparative study of dimensionality reduction techniques to enhance trace clustering performances

M. Song [a,*], H. Yang [a], S.H. Siadat [a], M. Pechenizkiy [b]

[a] School of Technology Management, Ulsan National Institute of Science and Technology, UNIST-GIL 50, 689-798 Ulsan, South Korea
[b] Department of Computer Science, Eindhoven University of Technology, Den Dolech 2, 5612 AZ Eindhoven, The Netherlands

ARTICLE INFO

Keywords:
Process mining
Trace clustering
Singular value decomposition
Random projection
PCA

ABSTRACT

Process mining techniques have been used to analyze event logs from information systems in order to derive useful patterns. However, in the big data era, real-life event logs are huge, unstructured, and complex so that traditional process mining techniques have difficulties in the analysis of big logs. To reduce the complexity during the analysis, trace clustering can be used to group similar traces together and to mine more structured and simpler process models for each of the clusters locally. However, a high dimensionality of the feature space in which all the traces are presented poses different problems to trace clustering. In this paper, we study the effect of applying dimensionality reduction (preprocessing) techniques on the performance of trace clustering. In our experimental study we use three popular feature transformation techniques; singular value decomposition (SVD), random projection (RP), and principal components analysis (PCA), and the state-of-the art trace clustering in process mining. The experimental results on the dataset constructed from a real event log recorded from patient treatment processes in a Dutch hospital show that dimensionality reduction can improve trace clustering performance with respect to the computation time and average fitness of the mined local process models.

1. Introduction

In order to make operational processes competitive, organizations continuously improve their processes. The starting point of process improvement is to understand processes to be improved. One of the techniques that can be used to understand business processes is process mining. Process mining aims at extracting useful information about business processes by analyzing event logs that contains process execution results (Gűnther & van der Aalst, 2007; van der Aalst, Weijters, & Maruster, 2004). Process mining results usually contain several information about processes such as process models, business performance metrics, organizational models, organizational relations, performance characteristics, etc. (Gűnther & van der Aalst, 2007; Maruster & Beest, 2009; Song & van der Aalst, 2008; van der Aalst et al., 2007). Recently, several organizations such as high-tech companies, hospitals, and municipalities utilize process mining techniques to improve their processes (Lemos, Sabino, Lima, & Oliveira, 2011; Mans, Schonenberg, Song, van der Aalst, & Bakker, 2008; Reijers, Song, & Jeong, 2009; Rozinat, Jong, Günther, & van der Aalst, 2009; Song, Gunther, & van der Aalst, 2008; van der Aalst et al., 2007).

However, traditional process mining techniques have difficulties in the analysis of "big" event logs, especially event logs from unstructured processes (e.g. health-care processes). Real-life event logs are huge, unstructured, and complex so that process mining results are often complicated and difficult to understand. The diversity of processes, i.e. each case has different kind of activities as well as different sequences of activities, cause complex process models. An example of a spaghetti-like process model is illustrated in Fig. 1(a).

To derive understandable process models, the trace clustering technique can be used to group similar cases into homogeneous subsets (clusters) according to their log traces (Song et al., 2008). Since cases in the same subset (cluster) have similar traces, the process model of each cluster (Fig. 1(b)) is more concise and understandable compare to the process model from the entire event log.

Despite the usefulness of the trace clustering technique, it is inclined to require a lot of computation time due to lots of features that most real-life event logs contain. Furthermore, many features in event logs are too trivial to be considered as features and unimportant features cause poor quality of trace clustering results. Thus, in this paper we apply dimensionality reduction (preprocessing) techniques to cope with the curse of dimensionality in trace clustering.

To apply dimensionality reduction techniques, a preprocessing step is added in the existing trace clustering procedure. Among

* Corresponding author. Tel.: +82 52 217 3116.
E-mail address: msong@unist.ac.kr (M. Song).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.12.078

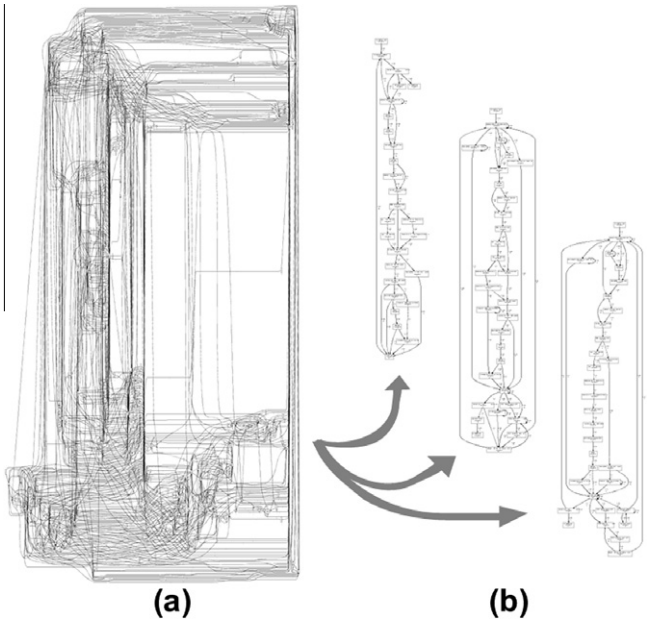


Fig. 1. An example of process model outcomes of the trace clustering.

many existing dimensionality reduction techniques, we apply singular value decomposition (SVD), random projection (RP), and principal components analysis (PCA). We conduct experiments to study the effect of dimensionality reduction on trace clustering. In the experimental study, a patient treatment process of a Dutch hospital is used and three measures such as average fitness of the locally mined process models, trace clustering processing time, and similarity are used as quality evaluation criteria. The results show that the trace clustering with dimensionality reduction techniques provides better performance in terms of computation time and average fitness.

This paper is organized as follows. Section 2 discusses related work and Section 3 introduces trace clustering and dimensionality reduction techniques used in this paper. Then, Section 4 describes our research framework which includes the research method, experiment procedures and settings, and evaluation criteria. Section 5 presents evaluation results, and finally Section 6 concludes the paper.

2. Related work

2.1. Process mining

The main idea of process mining is extracting valuable knowledge from event logs which are records of business executions (van der Aalst et al., 2004, 2007). An event log consists of events or 'audit trail entries', and each event refer to an activity for a specific case or process instance. Also each event contains information about the originator ("who executed the event") and time stamp ("when the event is executed") of the event (van der Aalst & de Medeiros, 2005). Recently, process mining techniques have been receiving more attention among researcher and practitioners, while applicability of process mining has been reported in various case studies. Process mining can be applied to event logs of various organizations such as public institutions (van der Aalst et al., 2007), manufacturers (Rozinat et al., 2009), telecom companies (Goedertier, Weerdt, Martens, Vanthienen, & Baesens, 2011), healthcare institutions (Mans et al., 2008). Moreover, it can be applied for internal fraud mitigation of organizations (Jans, van der Werf, Lybaert, & Vanhoof, 2011).

There exist three conceptual classes of process mining techniques which are discovery, conformance, and extension (Rozinat & van der Aalst, 2008). The concept of discovery aims at creating process models automatically from event logs (Jans et al., 2011; Rozinat & van der Aalst, 2008; Tsai, Jen, & Chen, 2010). In general, it is not trivial to obtain a process model which describes the event log perfectly. Thus, a wide range of techniques are developed for discovering process models from real-life event logs e.g. the alpha algorithm (de Medeiros, van der Aalst, & Weijters, 2003; van der Aalst et al., 2004), the heuristic miner (Weijters, van der Aalst, & de Medeiros, 2006), the fuzzy miner (Gűnther et al., 2007), and the genetic miner (de Medeiros & Weijters, 2005). The concept of conformance is about checking whether an existing process model matches a corresponding log. With this regards, conformance checking measures such as fitness and appropriateness have been developed (Jagadeesh Chandra Bose & van der Aalst, 2009; Rozinat & van der Aalst, 2008). The concept of extension aims at projecting information acquired from the event log into the process model (Maruster & Beest, 2009; Rozinat & van der Aalst, 2008).

2.2. Trace clustering

Despite the ability of many process mining algorithms in discovering process models from event logs, the resulting process models are usually spaghetti-like and too complex to understand. This situation is more feasible for less structured processes e.g. in healthcare domain due to the characteristics of the healthcare processes. With this regards, clustering techniques can be used for handling event logs by breaking down a log into several groups of clusters with similar types. Process mining can then extract simpler process models from each cluster. Trace clustering has been successfully applied in process mining and addressed in many literatures.

(Greco, Guzzo, Pontieri, & Sacca, 2006) used trace clustering to classify cases of event logs to facilitate the process of discovering expressive process models. In (Greco et al., 2006), a vector space model based on activities and transitions is used to find out appropriate clusters. (Song et al., 2008) proposed an approach to create profiles of logs with control-flow perspective, organization perspective, and data perspective. The items included in profiles are used as features which are the criteria of clustering algorithms. Therefore, (Song et al., 2008) derives clusters not only based on activities and transitions, but also based on originators, data and performance. Moreover, context-aware trace clustering based on a generic edit distance framework is studied in (Jagadeesh Chandra Bose & van der Aalst, 2009). Since the edit distance framework is very sensitive to the costs of edit operations, the authors proposed a method which automatically calculates the edit operations cost in order to handle the sensitivity of the cost function. The authors concluded that their proposed approach can improve the process mining result in comparison to other current trace clustering approaches with respect to fitness and comprehensibility.

The application of clustering techniques to facilitate process mining results in particular in healthcare domain has been also investigated in some literatures (Jagadeesh Chandra Bose & van der Aalst, 2009; Mans et al., 2008; Rebuge & Ferreira, 2012). (Mans et al., 2008) applied SOM (self-organizing map) algorithm to cluster a hospital event log and afterwards used a Heuristic Miner to generate more understandable process models. In (Rebuge & Ferreira, 2012), sequence clustering is used for business process analysis in healthcare domain. Trace clustering performs distance-base clustering techniques based on features of traces (Song et al., 2008), whereas sequence clustering performs based on sequential behavior of traces (Veiga & Ferreira, 2009).

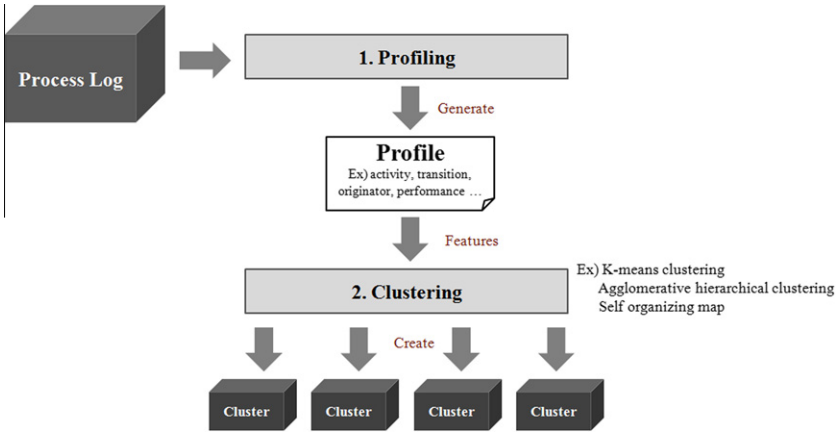


Fig. 2. Process of the trace clustering.

Process Log

Case ID	Log
1	(A, Park) (B, Kim) (C, Lee) (D, Choi) (E, An)
2	(A, Park) (B, Kim) (E, Choi)
3	(A, Park) (D, Kim) (F, Park)
⋮	⋮

Transition Profile

Case ID	AB	BC	CD	DE	EF	⋯
1	1	1	1	1	0	⋯
2	1	0	0	0	0	⋯
3	0	0	0	0	0	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Activity Profile

Case ID	A	B	C	D	E	F	⋯
1	1	1	1	1	1	0	⋯
2	1	1	0	0	1	0	⋯
3	1	0	0	1	0	1	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋱

Originator Profile

Case ID	Park	Kim	Lee	Choi	An	⋯
1	1	1	1	1	1	⋯
2	1	1	0	1	0	⋯
3	2	1	0	0	0	⋯
⋮	⋮	⋮	⋮	⋮	⋮	⋱

Fig. 3. The example of the trace profiles.

The above works are valuable lessons in applying clustering techniques to improve process mining results; however, there is a lack of research in performing a comparison study with the application of various clustering techniques in process mining. Generally, all clustering techniques are important methods in the data mining field (Jain & Dubes, 1988) and worth to be studied for the purpose of process mining. However, the clustering techniques applied to trace clustering in this paper are namely *K* means clustering, agglomerative hierarchical clustering and self-organizing map which are the most popular clustering algorithms in the data mining field.

2.3. Dimensionality reduction techniques

Dimensionality of the data means the number of attributes which describe every record in data. In data mining field, dimension reduction is an important issue in processing high-dimensional data (Bartl, Rezanková, & Sobisek, 2011; Zhang, Jiang, & Ye, 2010). Principal component analysis (PCA) and factor analysis (FA) are widely used dimensionality reduction techniques studied in many literatures (Bartl et al., 2011; Megalooikonomou, Li, & Wang, 2008; Tan, Steinbach, & Kumar, 2006; Xu & Wang, 2005). Categorical principal component analysis (CATPCA) is a dimensionality reduction technique that can be used when the attributes of data need to be transformed from categorical to quantitative attributes (Bartl et al., 2011). Multidimensional scaling (MDS) is a generalized technique of FA that can be used for dimension reduction ad explores similarities or dissimilarities in data (Bécavin, Tchitchek, Mintsa-Eya, Lesne, & Benecke, 2011; Cil, 2012). Moreover, other dimensionality reduction techniques such as random projection (Achlioptas, 2003; Bingham & Mannila, 2001; Johnson & Lindenstrauss, 1984), singular value decomposition (SVD) (Golub & Reinsch, 1970; Gong & Liu, 2000; Ma, Parhi, & Deprettere, 2001), and fisher discriminant analysis (Zhang et al., 2010) are developed and applied in literatures.

In the data mining field, as the methods of collecting data are developing, the features used to cluster data become much bigger while many of them are irrelevant and redundant. Therefore, the dimensionality reduction techniques are proposed to deal with irrelevant and redundant features. Despite the benefits of dimensionality reduction techniques, the application of these techniques is fairly unexplored for process mining. Among many preprocessing techniques, we use singular value decomposition, random

Table 1
Terms for distance measures.

Term	Explanation
c_j	Corresponds to the vector $<i_{j1}, i_{j2}, \ldots, i_{jn}>$
i_{jk}	The number of appearance of item k in the case j
k	kth item (feature or activity)
j	jth case
n	The number of features extracted from event log to be criteria of clustering algorithm

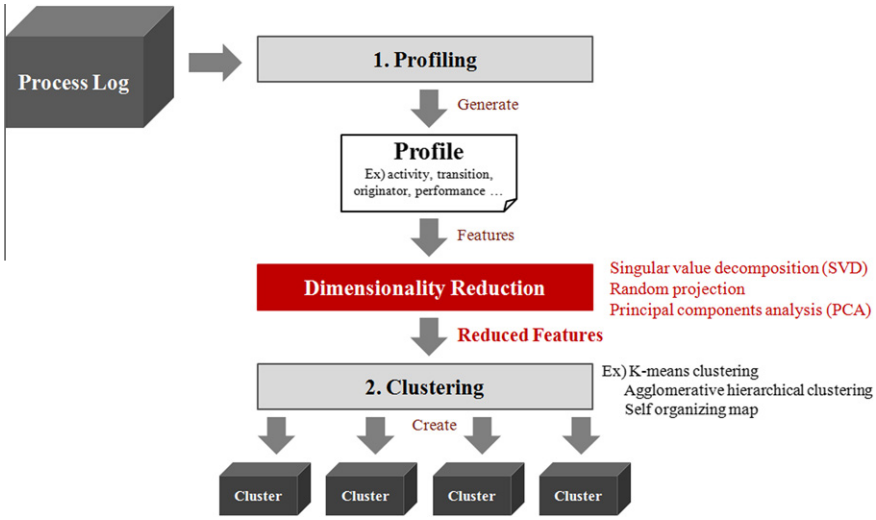


Fig. 4. Process of the trace clustering with application of dimensionality reduction technique.

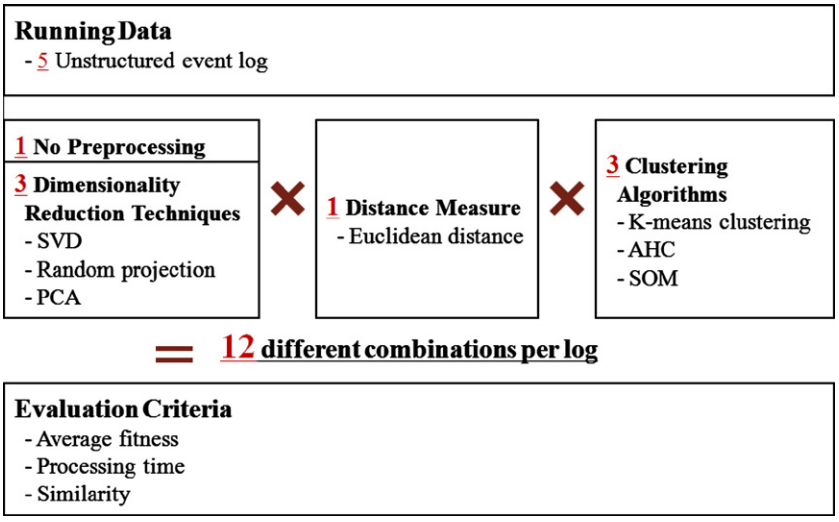


Fig. 5. Design of the experiments.

projection and principal components analysis in this paper. In the following we address related work on these techniques in data mining fields.

Random projection has been applied to various data sources such as text data, image data (Bingham et al., 2001), and cancellable biometrics in face recognition (Ying & Jin, 2007), in order to reduce the dimension of data. SVD is an excellent and powerful technique in many fields. For example, it can be implemented in signal modeling, system identification, image reconstruction, realization and reliable computations (Ma et al., 2001). In experiments with actual data, however, the result of separation by size of the singular values are usually not clear. Therefore, determining the number of singular values is very important. An appropriate singular value improves stability of the experiment and lowers the possibility of losing significant signal information (Sano, 1993). SVD has been used in many fields such as text retrieval (Nicholas & Dahlberg, 1998), video summarization (Gong et al., 2000), and hand gesture recognition (Liu & Kavakli, 2010). PCA uses clustering to predict user preferences (Goldberg, Roeder, Gupta, & Perkins, 2001). PCA has been reviewed and extended because of its potential applications. Categorical PCA and Nonlinear PCA are the extended versions of PCA, and they are being studied by many researchers (Meulman, van der Kooij, & Heiser, 2004).

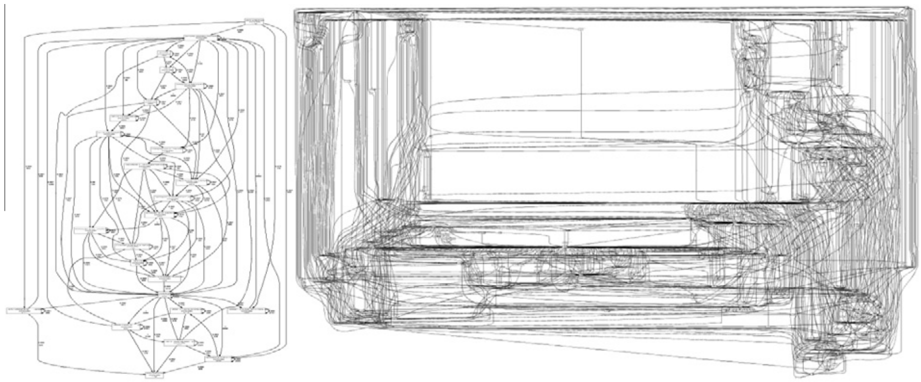
3. Trace Clustering and dimensionality reduction techniques

3.1. Trace clustering

Trace clustering classifies cases of a log into homogeneous subsets (clusters) according to features of the cases. Since cases in the same cluster are similar to each other, the process models of each

Table 2
The resulting logs after filtering.

Log name	Filtering (%)	# of events per case			# of types of events
		Min	Average	Max	
PL_1	1.0	3	18	25	25
PL_2	0.8	3	22	32	32
PL_3	0.5	3	28	48	49
PL_4	0.3	3	31	63	65
PL_5	0	1	33	113	624



(a)Process model of PL_1 (b)Process model of PL_5

Fig. 6. Process models of running data.

cluster are much simpler than the process models of the entire event log. Besides, by applying various process mining techniques to each cluster separately, we can extract useful information easier due to the simplicity of the logs from each cluster.

Fig. 2 illustrates that the process of trace clustering is divided into two phases namely profiling and clustering. In the profiling phase, a trace profile is generated. The features, which are items for comparing trace of each case, are organized in the trace profile. In the clustering phase, clustering algorithms are used to classify cases of the log. The clustering algorithms require a vector space to measure the distance between any two points which indicate cases in the log. Each axis of the vector space is corresponding to each feature of the trace profiles. In other words, the features of trace profile are used as criteria of the clustering algorithms in the clustering phase. In this paper, we use two trace profiles, which are activity and transition profiles and three clustering algorithms, which are *K* means clustering, agglomerative hierarchical clustering and self-organizing map. This section describes the trace profiles and the clustering algorithms used in this paper.

3.1.1. Trace profiles

All clustering algorithms require criteria for classifying datasets. In case of the trace clustering, the clustering algorithm uses log traces as classification criteria. The log traces are characterized in the format called trace profiles (Song et al., 2008). A trace profile consists of items that express trace of cases from a particular perspective, and every item in the trace profile can be used as a

No preprocessing

	Case ID
Cluster 1	1, 2, 3, 4
Cluster 2	5, 6, 7, 8
Cluster 3	9, 10, 11
Cluster 4	12, 13, 14

Preprocessed by SVD

	Case ID
Cluster 1	1, 2, 3
Cluster 2	4, 5, 6, 7, 8
Cluster 3	9, 10, 11, 12, 13
Cluster 4	14

(a) Case id composition of clusters

		Preprocessed by SVD			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
No preprocessing	Cluster 1	3	1	0	0
	Cluster 2	0	4	0	0
	Cluster 3	0	0	3	0
	Cluster 4	0	0	2	1

(b) A similarity matrix

		Preprocessed by SVD			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
No preprocessing	Cluster 1	3	1	0	0
	Cluster 2	0	4	0	0
	Cluster 3	0	0	3	0
	Cluster 4	0	0	2	1

		Preprocessed by SVD			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
No preprocessing	Cluster 1	3	1	0	0
	Cluster 2	0	4	0	0
	Cluster 3	0	0	3	0
	Cluster 4	0	0	2	1

		Preprocessed by SVD			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
No preprocessing	Cluster 1	3	1	0	0
	Cluster 2	0	4	0	0
	Cluster 3	0	0	3	0
	Cluster 4	0	0	2	1

		Preprocessed by SVD			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
No preprocessing	Cluster 1	3	1	0	0
	Cluster 2	0	4	0	0
	Cluster 3	0	0	3	0
	Cluster 4	0	0	2	1

(c) Processes for searching the highest total number of shared case IDs

		Preprocessed by SVD			
		Cluster 1	Cluster 2	Cluster 3	Cluster 4
No preprocessing	Cluster 1	3	1	0	0
	Cluster 2	0	4	0	0
	Cluster 3	0	0	3	0
	Cluster 4	0	0	2	1

(d) Results of process in (c)

Fig. 7. An example of similarity calculation processes.

Table A1
Average fitness results (*K*-means clustering).

Log name	K	No preprocessing	SVD	Random projection	PCA
PL_1	5	0.00104	0.20326	0.25792	0.00205
	6	0.00120	0.21747	0.23837	0.00197
	7	0.00829	0.21229	0.20589	0.00834
	8	0.00824	0.19591	0.20397	0.00808
	9	0.00699	0.19564	0.19880	0.00826
	10	0.00640	0.19591	0.19755	0.00806
PL_2	5	0.00000	0.20030	0.19515	0.00432
	6	0.00300	0.19104	0.18584	0.00427
	7	0.00000	0.19433	0.18474	0.00953
	8	0.02700	0.17921	0.18600	0.00112
	9	0.02763	0.17904	0.17896	0.03561
	10	0.00319	0.17868	0.17879	0.03171
PL_3	5	0.00241	0.00384	0.00408	0.02542
	6	0.03709	0.00375	0.00407	0.03295
	7	0.03313	0.00369	0.00381	0.03295
	8	0.02521	0.00358	0.00378	0.03537
	9	0.02311	0.00356	0.00370	0.03537
	10	0.02813	0.00346	0.00364	0.03316
PL_4	5	0.00000	0.00000	0.00000	0.02850
	6	0.02116	0.00000	0.00000	0.02870
	7	0.03164	0.00000	0.00000	0.02870
	8	0.01873	0.00000	0.00216	0.02214
	9	0.01860	0.00000	0.00216	0.02213
	10	0.02180	0.00000	0.00216	0.01696
PL_5	5	0.00000	0.00000	0.00000	0.00000
	6	0.00000	0.00000	0.00000	0.00000
	7	0.00000	0.00000	0.00000	0.00000
	8	0.00000	0.00000	0.00000	0.00000
	9	0.00000	0.00000	0.00088	0.00000
	10	0.00000	0.00000	0.00088	0.00088

criterion for classifying cases in the clustering phase. Note that all values in the trace profile are expressed in numerical value.

Fig. 3 illustrates examples of the trace profiles. In Fig. 3, the event log is written in numerical order of case id, and each case has a few parentheses. In one parenthesis, an activity is indicated with an alphabet, and the person who conducted the activity is recorded with his/her last name. Moreover, the order of parentheses shows the sequence of conducted activities. In the activity profile, each number in the profile means the number of each activity conducted in each case, and one activity is defined as one item. The transition profile shows the number of transition from one activity to another activity in each case. The originator profile is created in a similar way which shows the originators who are the workers in the event log. Therefore, information of each row is the profile vector of a trace in the log.

3.1.2. Distance measures

To classify cases into clusters, the clustering algorithms need a method to calculate the dissimilarities between cases. The cases can be projected in vector space based on the data in profiles. The method to calculate distances between cases of the event log is called 'distance measures'. There are many kinds of distance measures such as hamming distance, jaccard index (Tan et al., 2006), and correlation coefficient (Song et al., 2008), and they are usually originated from data mining field. In this paper, we use Euclidean distance (Duda, Hart, & Stork, 2000) to measure the dissimilarities between cases of event logs.

Through the profiles which are generated in the first phase of the trace clustering, we can project cases of the log to an *n*-dimensional vector space where *n* specifies the number of features extracted from the event log. Terms of distance measures are explained in Table 1. The Euclidean distance is defined as follows (Duda et al., 2000):

$$(c_j, C_k) = \sqrt{\sum_{i=1}^{n} ||i_{j,i} - i_{kl}||^2} \quad (1)$$

The Euclidean distance is used for computing the similarity between two vectors; it can calculate the similarity efficiently between two vectors regardless of the dimension of the vector space (Jeong, Kim, Kim, & Choi, 2006). However, the required time to compute the Euclidean distance between two high dimensional vectors is quite long. If we can identify the features that are trivial to be considered as features, we can reduce the total calculating time significantly by reducing the dimension of the vector space.

3.1.3. Clustering techniques

K means clustering algorithm is a frequently used partitioning method in practice (MacQueen, 1967; Song et al., 2008). By employing *K* means clustering, we can obtain *K* clusters by dividing the data into K groups from an event log. Even though multiple iterations are required to run the data, *K* means clustering algorithm is very efficient algorithm in comparison to other clustering algorithms (Pelleg & Moore, 2000). Many variations of *K* means clustering such as *X*-means clustering (Pelleg et al., 2000) and *K* harmonic means clustering have been constructed and studied to obtain better clustering results.

Agglomerative hierarchical clustering (AHC) is considered as one of the important clustering technique in the data mining field, since it has been studied relatively a long time in comparison to other kinds of clustering techniques (Tan et al., 2006). AHC algorithm starts with considering each point as a single cluster. Then clusters are merged according to distances between each cluster, and the same process is repeated until the number of clusters reaches to one (Zho & Karypis, 2005). AHC algorithm runs only once and creates a dendrogram which is a tree like diagram. In a dendrogram the height of each node indicates proportional intergroup dissimilarity between two daughters of the cluster (Witten, Frank, & Hall, 2011).

Self-organizing map (SOM) is a data clustering and visualization technique which is developed based on neural network analysis. SOM is useful to map high dimensional process data into low dimensional space which is much easier to analyze the event logs (Sarwar, Karypis, Konstan, & Riedl, 2000; Song et al., 2008; Tan et al., 2006). The goal of using SOM is clustering similar cases together and visualizing the result using colors and nodes.

3.2. Dimensionality reduction techniques

3.2.1. Singular value decomposition (SVD)

SVD is a technique for matrices dimensionality reduction and it can improve the scalability of Collaborative Filtering (CF) systems (Sarwar et al., 2000). Equation of SVD is as follow,

$$M = U \sum V^* \quad (2)$$

where M is an $m \times n$ matrix which consists of real and complex numbers, and the entries of M are component of dataset. In this paper, each column represents the cases and each row represents the feature created by profiling. According to Eq. (2), M is decomposed to three matrices which are U, $\sum$, V^*. The matrix U denotes an $m \times m$ orthogonal transformation matrix, the matrix $\sum = diag(\sigma_1, \sigma_2, \ldots, \sigma_n)$ is an $m \times n$ diagonal matrix, and the $n \times n$ unitary matrix V^* denotes the conjugate transpose of the matrix V (Wall, Rechtsteiner, & Rocha, 2003). The diagonal entries (σ_i) of the matrix $\sum$ are non-negative values with descending order from upper left corner of the matrix, and they are known as singular values of M. Also, when a rank is r, the singular values are satisfied as follows (Gong et al., 2000):

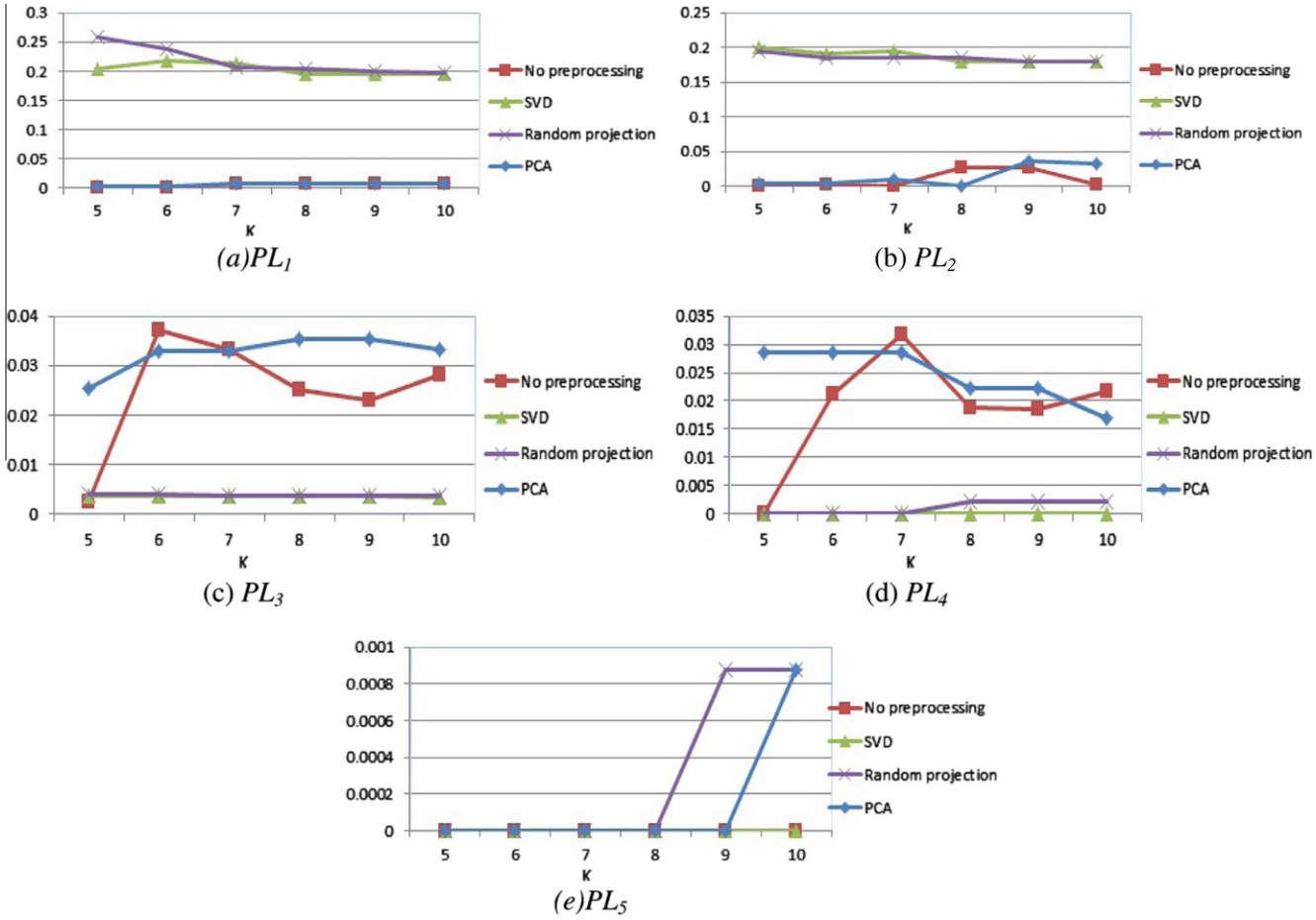


Fig. 8. The graphs of average fitness results (*K* means clustering).

$$\sigma_1 \leqslant \sigma_2 \ldots \leqslant \sigma_r \leqslant \sigma_{r+1} = \ldots = \sigma_n = 0 \tag{3}$$

In this paper, by selecting *k*-largest singular values, we can project the data to *k* dimension space. The diagonal entries σ_i (where *i* is larger than *k*) are set to 0, and then reduced matrix M_k is calculated. Then, the data in the matrix M_k are projected to *k* dimension space.

3.2.2. Random projection

Random projection is a technique which projects a set of data points to a randomly chosen low-dimensional space. Its equation is as follow:

$$X^{RP}_{k\times N} = R_{k\times d} X_{d\times N} \tag{4}$$

When the data has *N* cases and *d* features, we can randomly select *k* features by using random projection. Also in the process of selection, we use a $k \times d$ matrix *R* whose columns have unit lengths. In other words, we reduce the number of the features by multiplying the matrix *R* to the original data matrix *X* (Bingham & Mannila, 2001). Random projection also preserves important properties of a set of the data points, and the properties can be the distances between pairs of data (Johnson & Lindenstrauss, 1984). Moreover, it is computationally very efficient and has very strong probabilistic foundations (Achlioptas, 2003).

3.2.3. Principal components analysis (PCA)

PCA is an eigenvalue decomposition of the data covariance matrix, and it is used for low-rank approximation which compares the data through a linear function of the variables (Markos, Vozalis, & Margaritis, 2010). PCA is a technique which is used to reduce the dimensionality of the data by measuring the correlation among many variables in terms of principal components. The principal components are obtained by calculating eigenvalue problem of covariance matrix *C* as follows:

$$Cvi = \lambda_i v_i \tag{5}$$

The matrix *C* is covariance matrix of vectors of the original data *X*, and λ_i s are the eigenvalues of the matrix *C*, and v_i s are the corresponding eigenvectors. Then, in order to reduce the dimensionality of the data, the *k* eigenvectors which corresponds to the *k* largest eigenvalues need to be computed (Xu et al., 2005). Let us consider $E_k = [v_1, v_2, v_3, \ldots, v_k]$ and $\Lambda = [\lambda_1, \lambda_2, \lambda_3, \ldots, \lambda_k]$, then we have C $E_k = E_k \Lambda$. Finally we can obtain the following equation:

$$X^{PCA} = E_k^T X \tag{5}$$

According to the Eq. (5), the number of the features of the original data matrix *X* is reduced by multiplying with a $d \times k$ matrix E_k which has *k* eigenvectors corresponding to the *k* largest eigenvalues. The result matrix is X^{PCA} (Bingham & Mannila, 2001).

3.3. Extension of trace clustering

To incorporate with dimensionality reduction techniques, the trace clustering method is extended. There are a large number of features in the profiles of the event logs we use to test our experiments. However, using all the features as criteria for the clustering algorithms is too computationally expensive. Furthermore, some of the features should not be used as criteria for the clustering

Table A2
Average fitness results (AHC).

Log name	# of clusters	No preprocessing	SVD	Random projection	PCA
PL_1	5	0.00172	0.00172	0.00169	0.00174
	6	0.00172	0.00171	0.00169	0.00174
	7	0.00172	0.00171	0.00163	0.00173
	8	0.00172	0.00171	0.00163	0.00173
	9	0.00172	0.00171	0.00163	0.00173
	10	0.00172	0.21143	0.00163	0.00173
PL_2	5	0.00173	0.00174	0.08239	0.00174
	6	0.00173	0.00173	0.08239	0.00174
	7	0.00173	0.00173	0.08239	0.00174
	8	0.00173	0.00173	0.08016	0.00174
	9	0.00173	0.07500	0.08016	0.00174
	10	0.00173	0.07439	0.08016	0.00174
PL_3	5	0.00174	0.00173	0.00173	0.00174
	6	0.00174	0.00173	0.00107	0.00174
	7	0.00173	0.00172	0.00107	0.00173
	8	0.00173	0.00172	0.00107	0.00173
	9	0.00173	0.00172	0.00107	0.00173
	10	0.00173	0.00172	0.00107	0.00173
PL_4	5	0.00087	0.00087	0.00087	0.00087
	6	0.00087	0.00087	0.00000	0.00087
	7	0.00087	0.00087	0.00000	0.00087
	8	0.00087	0.00087	0.00000	0.00087
	9	0.00087	0.00087	0.00000	0.00087
	10	0.00086	0.00000	0.00000	0.00087
PL_5	5	0.00000	0.00000	0.00000	0.00000
	6	0.00000	0.00000	0.00000	0.00000
	7	0.00000	0.00000	0.00000	0.00000
	8	0.00000	0.00000	0.00000	0.00000
	9	0.00000	0.00000	0.00000	0.00000
	10	0.00000	0.00000	0.00000	0.00000

algorithms. To overcome the challenges of the trace clustering, we applied dimensionality reduction techniques to trace clustering as illustrated in Fig. 4. Therefore, as it is shown in Fig. 4, we can provide reduced number of the features to the clustering algorithms as clustering criteria.

4. Research framework

To investigate the relationships between the dimensionality reduction techniques and the clustering algorithms, several experiments were conducted. The design of the experiments is presented in Fig. 5. We used five real-life event logs for the experiments. They are unstructured event logs, and are basically similar hospital logs but have different complexities of log compositions. Details about the event logs are in Section 4.3. We used three dimensionality reduction techniques which are singular value decomposition (SVD), random projection, and principal components analysis (PCA) as well as three clustering algorithms which are *K* means clustering, agglomerative hierarchical clustering (AHC), and self-organizing map (SOM). Moreover, to estimate the influence of dimensionality reduction techniques to trace clustering results, we generated trace clustering results without performing preprocessing.

The cases can be projected in vector space based on the data in profiles. The distance between cases in the vector space is interpreted as the dissimilarity of the cases. We used Euclidean distance as the distance measure of the experiments. As illustrated in Fig. 5, each combination is composed of Euclidean distance measure, a clustering algorithm, and a dimensionality reduction technique. We designed the experiments to compare trace clustering results

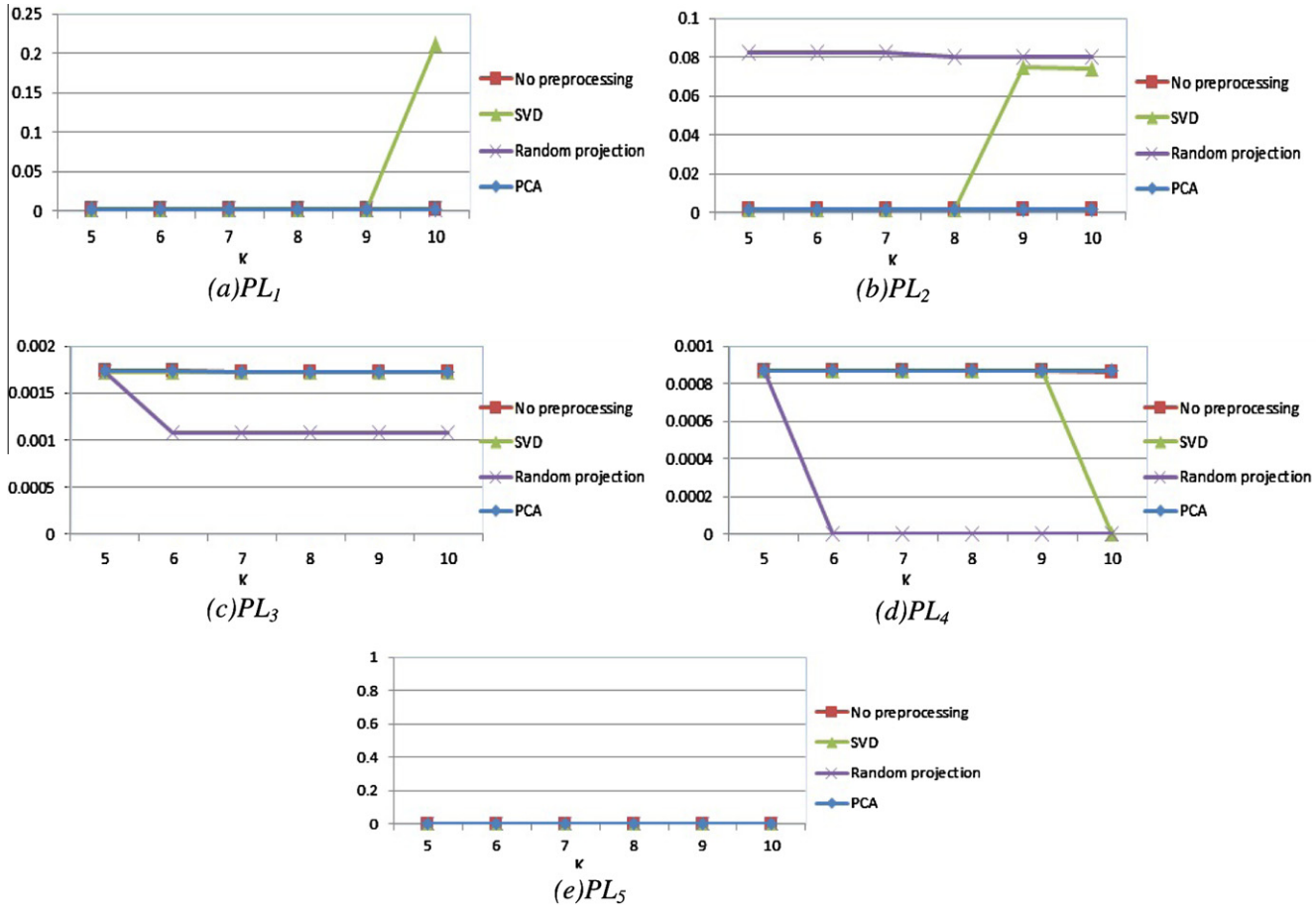


Fig. 9. The graphs of average fitness results (AHC).

Table 3
Average fitness results (SOM).

Log name	No preprocessing	SVD	Random projection	PCA
PL_1	0.11087	0.00175	0.18276	0.03398
PL_2	0.11365	0.00175	0.16271	0.13972
PL_3	0.00389	0.00175	0.00263	0.00000
PL_4	0.00400	0.00087	0.00000	0.00957
PL_5	0.00263	0.00088	0.00000	0.00000

of 12 combinations according to three evaluation criteria. Details about the evaluation criteria are in Section 4.4.

4.1. Experiment procedure

The process of the experiments is as follows. First, we implement the trace clustering to the experimental logs and achieve trace clustering results without preprocessing as control variables. Since we want to measure the size of effects caused by applying the dimensionality reduction techniques to trace clustering, we need the reference trace clustering results which do not affected by any kind of variables. Second, we implement the trace clustering with Euclidean distance, applying one of the preprocessing techniques and one of the clustering algorithms. Since there are three clustering algorithms and three preprocessing techniques, we can derive nine different trace clustering results for each log. In total, we can derive 12 different results per log including the control variable results. Finally, we compare and evaluate the outcomes. The comparison should be executed among the results that use the same clustering algorithm. In other words, results from the *K* means clustering, AHC and SOM should be analyzed separately.

4.2. Experiment setups

All the results are obtained using an Intel(R) Core(TM) i3 CPU 550 running at 3.20 GHz (4 CPUs) with 3072 MB RAM and Windows 7 Enterprise K 32-bit operating System. We use ProM 5.2 tool to test our experiments (Process Mining Group, 2009). ProM is an effective framework for performing process mining techniques which is able to analyze XES or MXML format event logs in a standard environment. Various kinds of plug-ins for process mining, analyzing, monitoring, and conversion have been developed in ProM and available for users.

4.3. Running data

We use extracted event log from the AMC hospital's databases to test our approach. The log is coming from a billing system of the hospital, and each event refers to a service delivered to a patient in 2005 and 2006. The event log is composed of 624 different event names, 1143 cases, and 150,291 events. In order to find out the influences of the log sizes to the experiment results, we set the log in five different sizes by using enhanced event log filter

Table 4
The best applicable dimensionality reduction techniques in terms of average fitness.

Log name	*K* means clustering	AHC	SOM
PL_1	SVD Random projection	SVD	Random projection
PL_2	SVD Random projection	Random projection	Random projection
PL_3	PCA	No preprocessing SVD PCA	No preprocessing
PL_4	PCA	PCA	PCA
PL_5	None	None	No preprocessing

provided in ProM. This way, we can remove events which are occurred less than a particular rate in the entire log, and generate filtered event log separately from the original event log.

Table 2 lists the resulting logs and their information after the filtering process. For example, the PL_2 log has 0.3% filtering that means the log does not contain events appeared less than 0.3% in the entire log. According to the number of events per case and types of events, PL_1 and PL_2 are considered simple logs whereas PL_3, PL_4 and PL_5 are relatively complex event logs. We apply Heuristic Miner (Weijters et al., 2006) in order to extract and gain insights about the control-flow perspective of process models from the hospital event logs. The heuristic miner algorithm can deal with noise and exceptions. Moreover, the combination of heuristic miner with clustering techniques has been investigated and concluded that it can provide satisfactory results regarding the characteristics of processes in healthcare domain (Jagadeesh Chandra Bose & van der Aalst, 2009; Mans et al., 2008; Rebuge & Ferreira, 2012).

Fig. 6 shows process models of PL_1 and PL_5 logs. It is easy to understand the complexities between two logs by comparing the models. Fig. 6(a) shows the process model generated based on PL_1 which is the simplest log, whereas Fig. 6(b) shows the process model generated based on PL_5 which is the unfiltered and most complex log. From an unstructured process model (e.g. the model in Fig. 6(b)), it is difficult to extract useful information because the model is too complex and containing too many activities and relations.

4.4. Evaluation criteria

The trace clustering results are achieved and analyzed according to three evaluation criteria namely the average fitness, the processing time, and the similarity.

4.4.1. Average fitness

The first evaluation criterion is the average fitness which is an average of fitness values derived from clusters generated by trace clustering. Fitness value explains how well an event log fits its process model. If the process model can regenerate traces of all cases

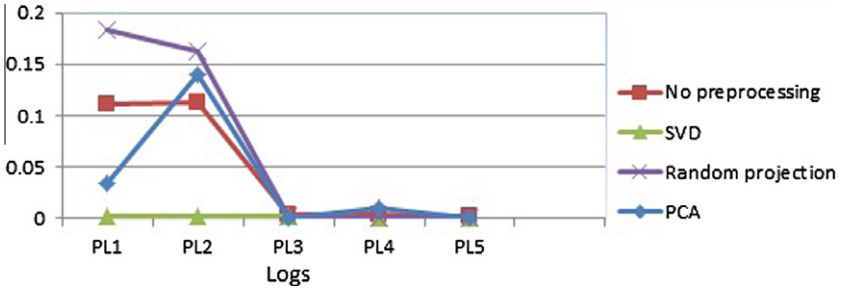


Fig. 10. The graph of average fitness results (SOM).

Table A3
Processing time results (*K*-means clustering).

Log name	K	No preprocessing	SVD	Random projection	PCA
PL_1	5	35.3	1.5	1.7	14.3
	6	44.9	1.5	1.8	17.1
	7	47.2	1.8	1.9	19.7
	8	56.3	2.0	2.7	22.5
	9	64.0	2.4	3.0	25.0
	10	68.2	2.6	3.1	29.8
PL_2	5	73.4	1.4	2.0	23.5
	6	78.9	1.9	2.2	26.7
	7	82.1	2.5	2.3	28.4
	8	95.2	2.9	2.6	34.2
	9	98.7	3.4	2.7	37.2
	10	116.2	4.2	3.0	40.6
PL_3	5	133.4	1.5	1.5	33.6
	6	151.5	1.6	1.7	38.5
	7	158.7	2.1	2.5	40.2
	8	187.5	2.6	2.8	44.5
	9	206.8	2.8	2.9	53.2
	10	225.1	4.0	3.2	74.5
PL_4	5	216.3	1.5	1.9	42.5
	6	223.8	1.6	2.6	49.4
	7	248.4	2.1	3.2	58.6
	8	289.1	2.9	4.2	68.1
	9	294.4	3.8	3.4	73.0
	10	317.0	4.1	3.7	81.3
PL_5	5	1798.3	2.3	1.9	142.4
	6	2156.6	2.7	2.3	184.0
	7	2298.8	2.9	2.9	202.3
	8	2640.7	3.3	3.6	240.7
	9	2718.8	3.6	4.1	280.5
	10	3035.8	3.9	4.9	298.8

in the log, we can say that the log fits the process model (Rozinat & van der Aalst, 2008). According to Rozinat and van der Aalst, in order to calculate fitness all cases of the log should be replayed in the process model which is called Petri net. While all cases of the log are replayed in the Petri net, we need to count the number of tokens according to their conditions. The token is consumed when each event is executed in the process model. The details about the token and Petri net are in (Rozinat & van der Aalst, 2008) and (de Medeiros et al., 2003). After counting tokens, the fitness can be achieved according to the following equation (Rozinat & van der Aalst, 2008):

$$fitness = \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} m_i}{\sum_{i=1}^{k} c_i}\right) + \frac{1}{2}\left(1 - \frac{\sum_{i=1}^{k} r_i}{\sum_{i=1}^{k} p_i}\right) \quad (6)$$

In the Eq. (6), k expresses the number of cases, m_i is the number of missing tokens, c_i indicates the number of consumed tokens, r_i indicates the number of remaining tokens, and p_i indicates the number of produced token. The resulted fitness value means how well a process model explains the event log. Therefore, if all cases are replayed perfectly without missing and remaining token, the fitness should be 1. In our experiments, we measured the fitness of each cluster and calculated the average of all fitness values, so we used the term 'average fitness'. The trace clustering result with a combination that shows the highest average fitness value is considered the best combination of clustering algorithm and dimensionality reduction technique.

4.4.2. Processing time

The second evaluation criterion is the processing time which shows the time to produce trace clustering results and it is required to be measured to show the efficiency of the dimensionality reduction techniques. By comparing the processing time of the trace clustering with the dimensionality reduction techniques, the effect of applying the preprocessing on the trace clustering can be explained. The trace clustering result with a combination that shows the shortest processing time is considered the best combination of clustering algorithm and dimensionality reduction technique.

4.4.3. Similarity

The third evaluation criterion is the similarity. The similarity is calculated with the object of observing the degree of change in trace clustering results while applying dimensionality reduction techniques. We compared the composition of clusters between control variable results and other results by calculating the rate of match between them. The rate of match is computed by comparing case IDs that each cluster contains.

Fig. 7 shows an example of the similarity calculation processes. In the example, we compared the trace clustering results without preprocessing and the trace clustering results preprocessed by SVD. First, we need to obtain the case IDs of each cluster in both results as shown in Fig. 7(a). Then, we generate a similarity matrix as depicted in Fig. 7(b). Next, we need to find out the maximum value of the entire values in the similarity matrix. Then, erase other values that belong to the same row and column of maximum value to compare clusters of two trace clustering results with satisfying one-to-one correspondence. If the maximum value exists more than once, we should choose the value which does not have the next highest value in the same row or column. The processes of searching the highest total number of shared case IDs are shown in Fig. 7(c). Note that values in blank show the number of case IDs that both clusters contain identically.

Therefore, through the processes in Fig. 7(c), we can obtain the highest total number of shared case IDs when clusters of two results are put in a one-to-one correspondence. Fig. 7(d) shows the outcomes resulted from the process in Fig. 7(c). Finally, the similarity is calculated as the highest total number of shared case IDs divided by the total number of case IDs. Therefore, in this example, the similarity is $(3 + 4 + 3 + 1)/14 = 0.7857$.

5. Computational results and discussions

5.1. Average fitness results

The average fitness results using K means clustering with different preprocessing techniques are shown in Table A1 (see Appendix). To do a comparative analysis of the average fitness values, we draw graphs of the results as illustrated in Fig. 8. The graphs show the average fitness values of each log when we use K means clustering with different preprocessing techniques (see Table 2 for the information about logs). The horizontal axis of the graph represents the K value, and the vertical axis of the graph represents the average fitness value. In Fig. 8, it can be seen that when we apply the trace clustering to PL_1 and PL_2, the combinations of K means clustering with both random projection and SVD are the best combinations in terms of average fitness. The differences can be interpreted in terms of optimal k, however it is not the focus of this paper. Moreover, when we apply the trace clustering to PL_3 and PL_4, K means clustering (in general without preprocessing) and its combination with PCA show a better fitness. Therefore, we obtained the fact that the size and complexity of the logs can affect the results of experiments. Note that although we used filtering to reduce the complexity of the logs, the average fitness values are still low due to the huge complexity of the original hospital log.

The average fitness results in using AHC with different preprocessing techniques are listed in Table A2 (see Appendix). The graphs in Fig. 9 show the average fitness values of each log when we use AHC with different preprocessing techniques. The horizontal axis of the graph represents the number of clusters, and the

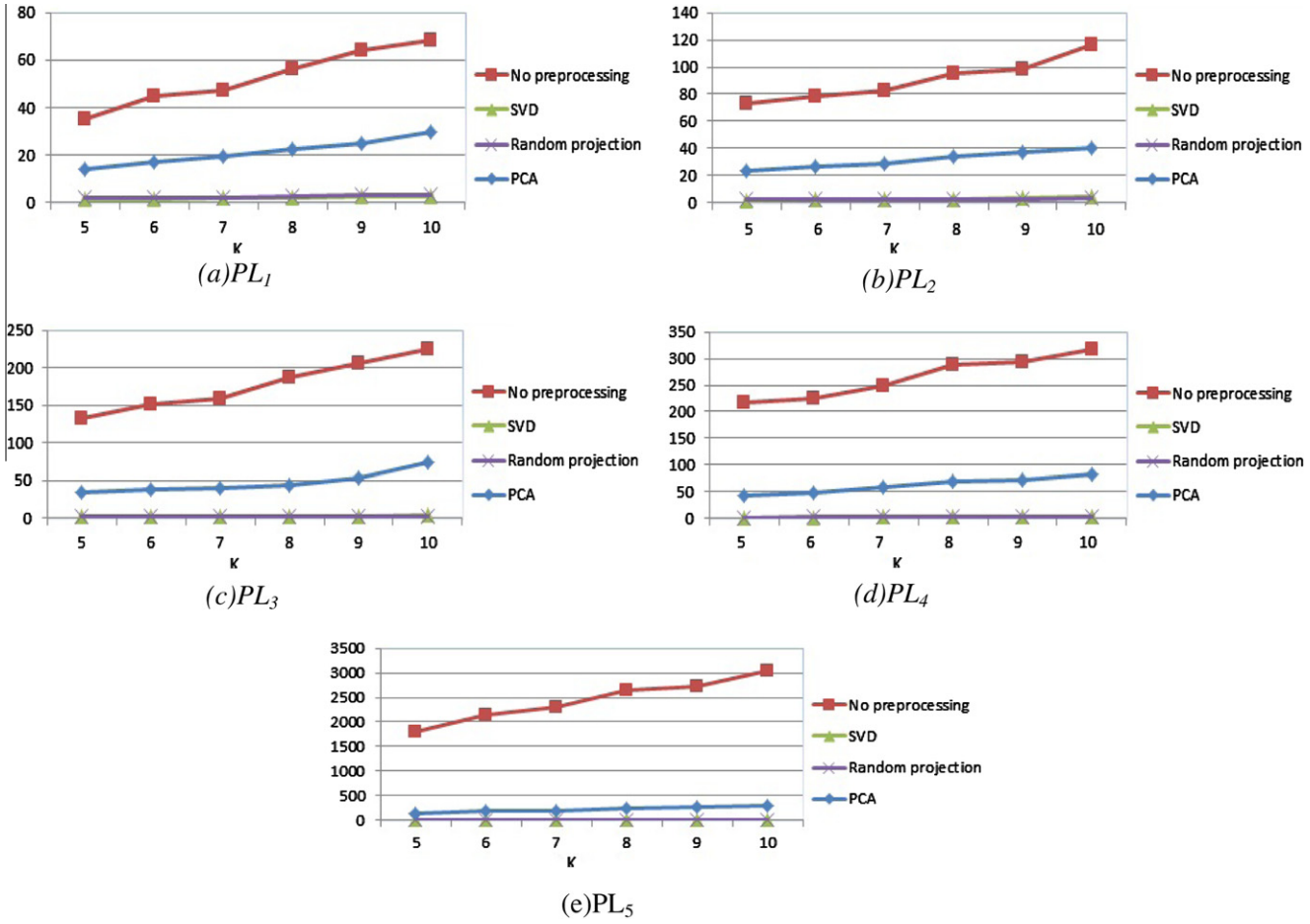


Fig. 11. The graphs of processing time results (*K*-means clustering).

vertical axis of the graph represents the average fitness value. As it can be seen in Fig. 9, the AHC technique cannot provide good fitness values with any preprocessing techniques and therefore is not suitable as a clustering technique for our case study.

The average fitness results in using SOM with different preprocessing techniques are listed in Table 3. Note that SOM does not require predetermined number of clusters. The graph in Fig. 10 shows the average fitness values when we use SOM with different preprocessing techniques. The horizontal axis of the graph represents the logs, and the vertical axis of the graph represents the average fitness value. As it can be seen in Fig. 10, the preprocessing techniques can improve the average fitness result of SOM only for simple event logs (e.g. PL_1 and PL_2). With this regards, the random projection can provide the best combination with SOM. The preprocessing techniques cannot make the fitness results any better for complex event logs as in PL_3, PL_4 and PL_5.

Table 4 lists the best dimensionality reduction techniques in terms of average fitness for each clustering algorithm according to the event logs. For example, it can be seen that PCA is the prominent dimensionality reduction technique to be applied for PL_3 and PL_4 logs, whereas SVD and random projection are better candidates for the PL_1 and PL_2 logs.

Overall, we can conclude that the combinations of *K* means clustering with random projection or SVD are the best for simple event logs (e.g. PL_1 and PL_2) while the combination of *K* means clustering with PCA can be suggested for complex event logs (e.g. PL_3 and PL_4). Note that none of the approaches is useful for PL_5 log since it was not filtered and this can demonstrate the importance of filtering in preparing the event logs for process mining.

Table 5
Processing time results (AHC).

Log name	No preprocessing	SVD	Random projection	PCA
PL_1	30.7	24.6	25.1	31.2
PL_2	39.4	29.9	30.8	41.4
PL_3	44.5	31.4	32.2	46.8
PL_4	56.9	36.5	36.2	58.2
PL_5	236.4	70.6	69.2	71.7

Nevertheless, the SOM clustering without preprocessing can be an option in this situation (the conclusion is achieved by comparing Table 3 with Table A1). In our case study, we realized that when no preprocessing is involve then SOM clustering has better performance in simple logs (e.g. PL_1 and PL_2) whereas *K* means clustering is more suitable for complex logs (e.g. PL_3 and PL_4) with respects to fitness values.

5.2. Processing time results

Table A3 (see Appendix) lists the processing time of the logs in using *K* means clustering with various preprocessing techniques. To do a comparative analysis of the processing time, the results are illustrated in Fig. 11. The graphs show the processing time of each log when we use *K* means clustering while applying different preprocessing techniques. The horizontal axis of the graph represents the *K* value, and the vertical axis of the graph represents the consumed processing time to cluster cases (in seconds). As it can be seen, SVD and random projection have the lowest processing time in all the logs.

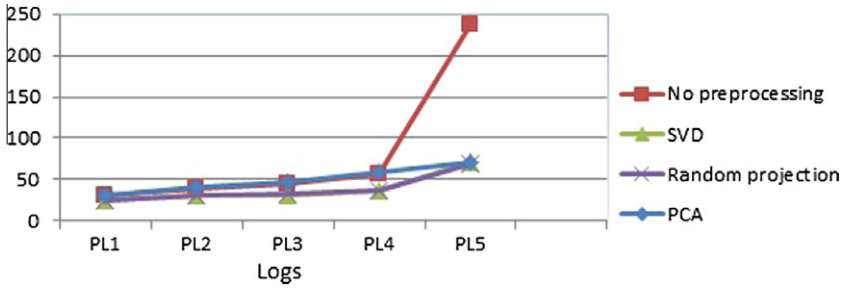


Fig. 12. The graph of processing time results (AHC).

Table 6
Processing time results (SOM).

Log name	No preprocessing	SVD	Random projection	PCA
PL_1	9.2	0.1	0.1	5.2
PL_2	18.0	0.1	0.1	7.9
PL_3	49.4	0.1	0.1	11.7
PL_4	117.0	0.1	0.1	22.1
PL_5	4796.0	0.1	0.1	97.1

Table 5 lists the results of processing time of the logs in using AHC with various preprocessing techniques. We achieved the same processing time for every cluster in each log in using AHC as clustering algorithm. Fig. 12 shows the processing time of each log when we use AHC while applying different preprocessing techniques. The horizontal axis of the graph represents name of the log, and the vertical axis of the graph represents the time-consumed to cluster cases (in seconds). As it can be seen, SVD and random projection techniques have the lowest processing time in all the logs and are the best candidates to be combined with AHC clustering with respect to processing time criterion. However, the processing times are still relatively high and therefore the AHC clustering is not recommended regarding processing time criterion.

Table 6 lists the processing time of the logs when we use SOM with various preprocessing techniques. Fig. 13 shows the graph for the processing time of each log in using SOM while applying different preprocessing techniques. The horizontal axis of the graph represents the logs, and the vertical axis of the graph represents the processing time of cluster cases (in seconds).

Table 7 lists the best dimensionality reduction techniques in terms of the processing time for each clustering algorithm with respect to the logs. According to the results, when we use trace clustering, it is better to apply SVD or random projection to decrease the clustering time. However, only the SOM can significantly reduce the processing times and therefore is the best candidate with respect to the processing time criterion.

5.3. Similarity results

We calculated the similarity criterion by comparing each result with its relevant control variable. Therefore, the 'No preprocessing' option does not exist in this analysis. The rates of match values in using *K* means clustering with different preprocessing techniques are calculated and listed in Table A4 (see Appendix). Fig. 14 show the similarity values of each log when we use *K* means clustering while applying different preprocessing techniques. The horizontal axis represents the *K* values and the vertical axis represents the similarity. It can be seen that PCA has the highest similarity and therefore it makes the best combination with *K* mean clustering.

Table 7
The best applicable dimensionality reduction techniques in terms of processing time.

Log name	*K* means clustering	AHC	SOM
PL_1	SVD Random projection	SVD Random projection	SVD Random projection
PL_2			
PL_3			
PL_4			
PL_5			

The rates of match values in using AHC with different preprocessing techniques are calculated and listed in Table A5 (see Appendix). Fig. 15 shows the graphs of the similarity values of each log when we use AHC while applying different preprocessing techniques. According to the graphs, PCA has the highest similarity and therefore it is the prominent preprocessing technique to be combined with AHC clustering. In addition, SVD shows a good similarity in PL_1 to PL_4 logs and can be another candidate to be used with the AHC clustering.

The rates of match values in using SOM with different preprocessing techniques are calculated and listed in Table 8. Fig. 16 shows the graph of the similarity values of each log listed in Table 8. The horizontal axis represents the number of clusters and the vertical axis represents the rate of match to control variable result. It can be seen that in most cases (e.g. except PL_2), SVD has the highest similarity.

Table 9 lists the dimensionality reduction techniques which have the highest similarity values for each clustering algorithm with respect to the event logs. According to the Table 9, the combinations of *K* means clustering with PCA, AHC with SVD and PCA, and finally SOM with SVD are the best candidates.

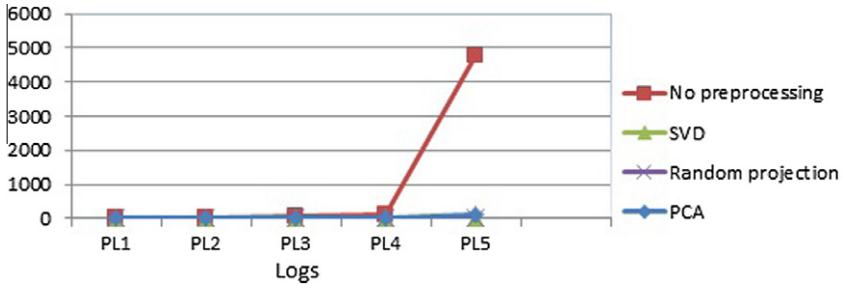


Fig. 13. The graph of processing time results (SOM).

Table A4
Similarity results (*K*-means clustering).

Log name	K	SVD	Random projection	PCA
PL_1	5	0.33050	0.31140	0.81080
	6	0.34970	0.27430	0.87660
	7	0.26470	0.22630	0.85390
	8	0.25390	0.22040	0.82280
	9	0.26710	0.22870	0.82630
	10	0.25750	0.23950	0.64550
PL_2	5	0.36520	0.38430	0.48880
	6	0.37530	0.37080	0.49780
	7	0.35510	0.35060	0.54270
	8	0.34940	0.32470	0.73820
	9	0.28540	0.27750	0.66520
	10	0.26400	0.26400	0.64160
PL_3	5	0.35556	0.32111	0.56222
	6	0.38444	0.32111	0.55444
	7	0.38111	0.35556	0.59444
	8	0.36556	0.35222	0.58333
	9	0.28111	0.26778	0.65667
	10	0.25444	0.22778	0.70000
PL_4	5	0.38770	0.44920	0.34340
	6	0.37260	0.32290	0.56050
	7	0.34560	0.30890	0.68030
	8	0.35960	0.31210	0.59400
	9	0.34770	0.29050	0.67060
	10	0.34670	0.28940	0.66090
PL_5	5	0.25980	0.80580	0.57390
	6	0.32020	0.65270	0.67280
	7	0.26950	0.64390	0.73320
	8	0.26600	0.60630	0.69820
	9	0.24850	0.52060	0.68070
	10	0.23710	0.51880	0.68850

Table A5
Similarity results (AHC).

Log name	# of Clusters	SVD	Random projection	PCA
PL_1	5	0.93290	0.91380	0.95810
	6	0.95210	0.90540	0.95330
	7	0.95570	0.80840	0.95570
	8	0.95570	0.80600	0.95570
	9	0.94370	0.80840	0.95330
	10	0.67070	0.80960	0.95210
PL_2	5	0.95730	0.93480	0.95840
	6	0.97980	0.94160	0.95960
	7	0.97750	0.94270	0.95960
	8	0.98090	0.89100	0.95960
	9	0.68090	0.88650	0.95840
	10	0.66180	0.88760	0.95730
PL_3	5	0.97556	0.94444	0.97889
	6	0.95778	0.62111	0.97778
	7	0.96667	0.62889	0.95444
	8	0.96889	0.62667	0.95333
	9	0.96667	0.62667	0.95556
	10	0.95667	0.63444	0.95556
PL_4	5	0.98270	0.98920	0.98490
	6	0.96110	0.69110	0.98600
	7	0.97950	0.69550	0.96440
	8	0.97300	0.69650	0.96540
	9	0.97080	0.69440	0.97520
	10	0.77860	0.70410	0.94380
PL_5	5	0.54420	0.96590	0.99300
	6	0.53280	0.96410	0.99300
	7	0.53280	0.92480	0.99480
	8	0.52060	0.93880	0.96760
	9	0.52230	0.93880	0.96680
	10	0.52230	0.93610	0.96410

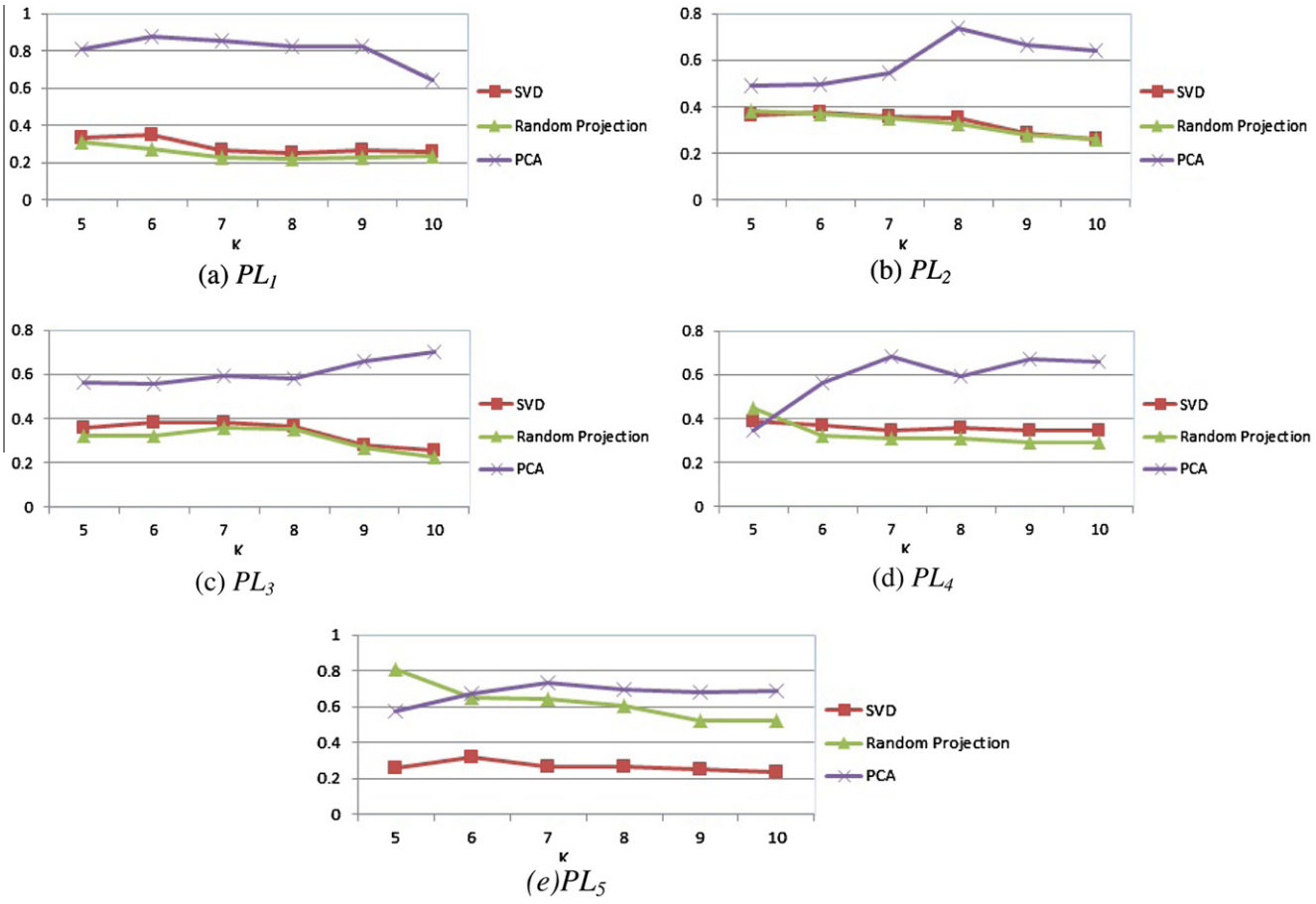


Fig. 14. The graphs of similarity results (*K*-means clustering).

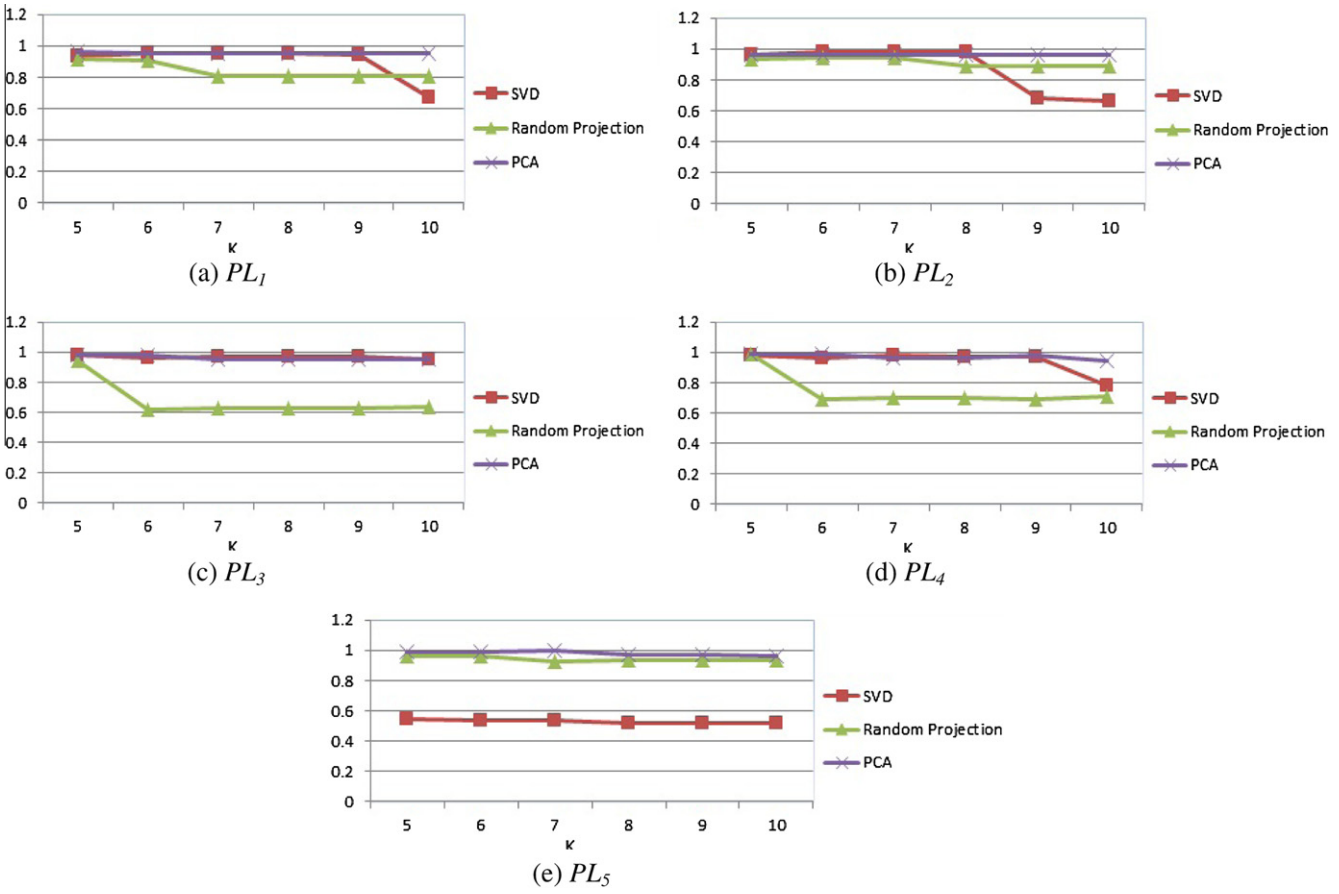


Fig. 15. The graphs of similarity results (AHC).

Table 8
Similarity results (SOM).

Log name	SVD	Random projection	PCA
PL_1	0.53770	0.39400	0.33290
PL_2	0.37870	0.43260	0.36400
PL_3	0.41556	0.35444	0.26667
PL_4	0.66630	0.37260	0.31750
PL_5	0.38320	0.30530	0.39460

Here we summarize the fitness and similarity results for *K* means and SOM clustering in Tables 10 and 11 respectively.

We concluded that the combinations of SVD or random projection with *K* means can substantially increase the fitness values for simple logs (see Table 10, Table A4 and Fig. 8). However, SVD and random projection do not have high similarity values in combination with *K* means clustering (see Table 9 and Table A4). This can be interpreted that since SVD and random projection considerably increase the fitness values, therefore they are able to reduce the noise and eventually their combination with *K* means clustering have a low similarity value. It can be seen in the Table 10 that PCA has the highest similarity in this case. On the other hand for complex logs e.g. PL_3 and PL_4, PCA is the best combination with *K* means clustering (see Table 10); however it only slightly increases the fitness values (see Table A4 and Fig. 8). Therefore the combination of PCA with *K* means clustering does not noticeably change the clustering results and it still can provide a reasonable similarity value in comparison to the other approaches. Regarding SOM clustering, although random projection for simple logs and no preprocessing for complex logs have the highest fitness values, however SVD can provide the best similarity in comparison to the other approaches (see Tables 8 and 11).

6. Conclusion

Process mining is an appropriate approach for extracting healthcare process models. However, due to the characteristics of

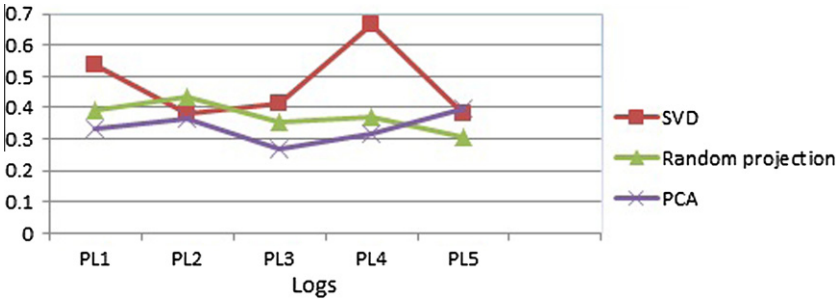


Fig. 16. The graph of similarity results (SOM).

Table 9
The combinations with the highest similarity values.

Log name	K means clustering	AHC	SOM
PL_1	PCA	PCA	SVD
PL_2		SVD	Random projection
PL_3		PCA	SVD
PL_4			
PL_5		PCA	SVD PCA

Table 10
The dimensionality reduction techniques for K means clustering in terms of fitness and similarity.

Log name	Fitness	Similarity
PL_1	SVD Random projection	PCA
PL_2		
PL_3		
PL_4	PCA	PCA

Table 11
The dimensionality reduction techniques for SOM clustering in terms of fitness and similarity.

Log name	Fitness	Similarity
PL_1, PL_2	Random projection	SVD
PL_3, PL_4	No preprocessing	SVD

healthcare processes, the process mining might results complex models that are difficult to understand. This motivates the efforts to perform trace clustering techniques to achieve more understandable process models. In this paper, in order to boast up the performances of trace clustering algorithms, we applied dimensionality reduction techniques. Therefore, we conducted various experiments with the patient treatment processes of a Dutch hospital to discover relationships and best combinations between the dimensionality reduction techniques and clustering algorithms.

We considered fitness value and processing time as the evaluation criteria of clustering results. According to the results, the average fitness value was improved by applying dimensionality reduction techniques to trace clustering. Moreover, processing time of trace clustering was effectively reduced with dimensionality reduction techniques. In other words, by applying dimensionality reduction techniques, we could improve trace clustering performances.

The main conclusions can be summarized as follows. First, the results of using dimensionality reduction techniques in terms of fitness could be various depending on the size and complexity of event logs as well as the applied clustering algorithms. We realized that the combination of *K* means clustering with SVD or random projection has the best fitness value for simple event logs, whereas the combination of *K* means with PCA is more suitable for complex event logs in our case study. Furthermore, when no preprocessing is used SOM performs better for simple logs whereas *K* means is preferred for complex event logs. Second, the results show that the preprocessing techniques are able to effectively reduce the required time for trace clustering processes. Among all dimensionality reduction techniques, SVD and random projection significantly decrease processing time for trace clustering regardless of complexity of the logs or type of the clustering algorithms. However, the combination of SOM with SVD or random projection showed the best processing time. Third, the dimensionality reduction techniques which result the highest similarity values are PCA for *K* means clustering, SVD and PCA for AHC, and SVD for SOM.

Acknowledgement

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2011-0010561).

References

Achlioptas, D. (2003). Database-friendly random projections: Johnson–Lindenstrauss with binary coins. *Journal of Computer and System Sciences, 66*(4), 671–687.

Bartl, E., Rezanková, H. & Sobisek, L. (2011). Comparison of classical dimensionality reduction methods with Novel approach based on formal concept analysis. In J. Yao, S. Ramanna, G. Wang, & Z. Suraj, (Eds.), *Rough sets and knowledge technology (RSKT 2011)*, October 9–12 2011, Banff, Canada. *Lecture notes in computer science* (Vol. 6954, pp. 26–35). Springer.

Bécavin, C., Tchitchek, N., Mintsa-Eya, C., Lesne, A., & Benecke, A. (2011). Improving the efficiency of multidimensional scaling in the analysis of high-dimensional data using singular value decomposition. *Bioinformatics, 27*(10), 1413–1421.

Bingham, E., & Mannila, H. (2001). Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on knowledge discovery and data mining (KDD 2001)*, August 26–29 2001, ACM: San Francisco, CA, USA. pp. 245–250.

Jagadeesh Chandra Bose, R. P. & van der Aalst, W. M. P. (2009). Context Aware Trace Clustering: Towards Improving Process Mining Results. In *Proceedings of the SIAM international conference on data mining (SDM 2009)*, April 30–May 2 2009. (pp. 401–412). Sparks, Nevada, USA.

Cil, I. (2012). Consumption universes based supermarket layout through association rule mining and multidimensional scaling. *Expert Systems with Applications, 39*(10), 8611–8625.

de Medeiros, A. K. A., van der Aalst, W. M. P., & Weijters, A. J. M. M. (2003). Workflow Mining: Current status and future directions. In: R. Meersman, Z. Tari, D. C. Schmidt, (Eds.), *On the move to meaningful internet systems 2003: CoopIS, DOA, and ODBASE – OTM confederated international conferences (CoopIS, DOA, and ODBASE 2003), November 3–7 2003. Catania, Sicily, Italy, Lecture notes in computer science* (Vol. 2888, pp. 389–406). Springer.

de Medeiros, A. K. A., & Weijters, A. J. M. M. (2005). Genetic process mining. *Lecture notes in computer science* (Vol. 3536, pp. 48–69). Springer.

Duda, R. O., Hart, P. E. & Stork, D. G. (2000). *Pattern classification* (2nd ed.). John Wiely and Sons: New York.

Goedertier, S., Weerdt, J. D., Martens, D., Vanthienen, J., & Baesens, B. (2011). Process discovery in event logs: An application in the telecom industry. *Applied Soft Computing, 11*(2), 1697–1710.

Goldberg, K., Roeder, T., Gupta, D., & Perkins, C. (2001). Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval Journal, 4*(2), 133–151.

Golub, G. H., & Reinsch, C. (1970). Singular value decomposition and least squares solution. *Numerische Mathematik, 14*(5), 403–420.

Gong, Y. & Liu, X. (2000). Video Summarization using Singular Value Decomposition. In *2000 conference on computer vision and pattern recognition (CVPR 2000)*, June 13–15 2000, (Vol. 1, pp. 174–180). Hilton Head, SC, USA: IEEE Computer Society.

Greco, G., Guzzo, A., Pontieri, L., & Sacca, D. (2006). Discovering expressive process models by clustering log traces. *IEEE Transactions on Knowledge and Data Engineering, 18*(8), 1010–1027.

Gűnther, C. W. & van der Aalst, W. M. P. (2007). Fuzzy Mining – Adaptive Process Simplification Based on Multi-Perspective Metrics. In G. Alonso, P. Dadam, & M. Rosemann (Eds.), *Business process management, 5th international conference (BPM 2007)*, September 24–28 2007, Brisbane, Australia: Proceedings. *Lecture notes in computer science* (Vol. 4714, pp. 328–343). Springer.

Jain, A. K., & Dubes, R. C. (1988). *Algorithms for clustering data.* Englewood Cliffs: Prentice-Hall Inc.

Jans, M., van der Werf, J. E. M., Lybaert, N., & Vanhoof, K. (2011). A business process mining application for internal transaction fraud mitigation. *Expert Systems with Applications, 38*(10), 13351–13359.

Jeong, S., Kim, S. W., Kim, K. & Choi, B. U. (2006). An effective method for approximating the euclidean distance in high-dimensional space. In S. Bressan, J. Küng, & R. Wagner (Eds.), *Database and expert systems applications 17th international conference (DEXA 2006)* September 4–8 2006. Kraków, Poland: Proceedings. *Lecture notes in computer science* (Vol. 4080, pp. 863–872). Springer.

Johnson, W. B., & Lindenstrauss, J. (1984). Extensions of lipshitz mapping into Hilbert space. *Contemporary Mathematics, 26*, 189–206.

Lemos, A. M., Sabino, C. C., Lima, R. M. F., & Oliveira, C. A. L. (2011). Using process mining in software development process management: A case study. In *Proceedings of the IEEE international conference on systems, man and cybernetics (SMC 2011).* October 9–12 2011. (pp. 1181–1186). Anchorage, Alaska, USA.

Liu, J., & Kavakli, M. (2010). Hand gesture recognition based on segmented singular value decomposition. In R. Setchi, I. Jordanov, R. J. Howlett, & L. C. Jain (Eds.), *Knowledge-based and intelligent information and engineering systems – 14th international conference (KES 2010)*, September 8–10 2010. (pp. 214–223) Cardiff, UK.

Ma, J., Parhi, K. K., & Deprettere, E. F. (2001). A unified algebraic transformation approach for parallel recursive and adaptive filtering and SVD algorithms. *IEEE Transactions on Signal Processing, 49*(2), 424–437.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observation. In *Proceedings of the 5th berkeley symp. on mathematical statistics and probability*. (pp. 281–297). University of California Press.

Mans, R. S., Schonenberg, M. H., Song, M., van der Aalst, W. M. P., & Bakker, P. J. M. (2008). Process mining in healthcare – a case study. In L. Azevedo & A. R. Londral (Eds.), *Proceedings of the first international conference on health informatics (HEALTHINF'08)*, January 28–31 2008. Funchal, Madeira, Portugal: Institute for Systems and Technologies of Information, *Control and communication*. (pp. 118–125). IEEE Computer Society.

Markos, A. I., Vozalis, M. G. & Margaritis, K. G. (2010). An optimal scaling approach to collaborative filtering using categorical principal component analysis and neighborhood formation. In H. Papadopoulos, A. S. Andreou, & M. Bramer (Eds.), *Artificial intelligence applications and innovations (AIAI 2010)*, October 6–7 2010. Larnaca, Cyprus: Proceedings. *IFIP Advances in information and communication technology* (Vol. 339, pp. 22–29). Springer.

Maruster, L., & Beest, N. R. T. P. (2009). Redesigning business processes: A methodology based on simulation and process mining techniques. *Knowledge Information Systems, 21*, 267–297. v.

Megalooikonomou, V., Li, G., & Wang, Q. (2008). A dimensionality reduction technique for efficient time series similarity analysis. *Information Systems, 33*(1), 115–132.

Meulman, J., van der Kooij, A., & Heiser, W. (2004). Principal components analysis with nonlinear optimal scaling transformations for ordinal and nominal data. In D. Kaplan (Ed.), *Handbook of quantitative methods in the social sciences*. Newbury Park: Sage Publications.

Nicholas, C. K., & Dahlberg, R. (1998). Spotting Topics with the Singular Value Decomposition. In E. V. Munson, C. K. Nicholas, & D. Wood (Eds.), *Principles of digital document processing, 4th International workshop (PODDP'98)*, March 29–30 1998. Saint Malo, France: Proceedings. *Lecture notes in computer science* (Vol. 1481, pp. 82–91). Springer.

Pelleg, D., & Moore, A. W. (2000). *X*-means: Extending *K* means with efficient estimation of the number of clusters. In P. Langley (Eds), *Proceedings of the seventeenth international conference on machine learning (ICML 2000)*, June 29–July 2, 2000. Stanford University: Stanford, CA, USA. (pp. 727–734). Morgan Kaufmann.

Process Mining Group, Math&CS department, Eindhoven University of Technology (2009). <http://www.processmining.org/prom/start>.

Rebuge, A., & Ferreira, D. R. (2012). Business process analysis in healthcare environments: A methodology based on process mining. *Information Systems, 37*(2), 99–116.

Reijers, H. A., Song, M., & Jeong, B. (2009). Analysis of a collaborative workflow process with distributed actors. *Information Systems Frontiers, 11*(3), 307–322.

Rozinat, A., Jong, I. S. M. d., Günther, C. W., & van der Aalst, W. M. P. (2009). Process mining applied to the test process of wafer scanners in ASML. *In IEEE Transactions on Systems, Man, and Cybernetics, Part C (RSMC), 39*, 474–479.

Rozinat, A., & van der Aalst, W. M. P. (2008). Conformance checking of processes based on monitoring real behavior. *Information Systems, 33*(1), 64–95.

Sano, A. (1993). Optimally regularized inverse of singular value decomposition and application to signal extrapolation. *Signal Processing, 30*, 163–176.

Sarwar, B. M., Karypis, G., Konstan, J. A. & Riedl, J. T. (2000). Application of dimensionality reduction in recommender systems – a case study. *ACM WebKDD 2000 web mining for E-commerce*, Workshop. (pp. 82–90).

Song, M., Gunther, C. W., & van der Aalst, W. M. P. (2008). Trace clustering in process mining. In D. Ardagna, M. Mecella, & J. Yang (Eds.), *Business process management workshops (BPM 2008)*, September 1–4 2008. Milano, Italy. *Lecture notes in business information processing* (Vol. 17, pp. 109–120). Springer.

Song, M., & van der Aalst, W. M. P. (2008). Towards comprehensive support for organizational mining. *Decision Support Systems, 46*(1), 300–317.

Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to data mining*. Boston, MA, USA: Pearson Addison Wesley.

Tsai, C.-Y., Jen, H., & Chen, I.-C. (2010). Time-interval process model discovery and validation – a genetic process mining approach. *Applied Intelligence, 33*(1), 54–66.

van der Aalst, W. M. P., & de Medeiros, A. K. A. (2005). Process mining and security: Detecting anomalous process executions and checking process conformance. *Electronic Notes in Theoretical Computer Science, 121*, 3–21.

van der Aalst, W. M. P., Reijers, H. A., Weijters, A. J. M. M., van Dongen, B. F., de Medeiros, A. K. A., Song, M., et al. (2007). Business process mining: an industrial application. *Information Systems, 32*(5), 713–732.

van der Aalst, W. M. P., Weijters, A. J. M. M., & Maruster, L. (2004). Workflow mining: Discovering process models from event logs. *IEEE Transactions on Knowledge and Data Engineering, 16*(9), 1128–1142.

Veiga, G. M., & Ferreira, D. R. (2009). Understanding spaghetti models with sequence clustering for ProM. In *Business Process Management Workshops (BPI 2009)* (pp. 92–103). Germany: Ulm.

Wall, M., Rechtsteiner, A., & Rocha, L. M. (2003). Singular value decomposition and principal component analysis. In I. D. P. Berrar, W. Dubitzky, & M. Granzow (Eds.), *A practical approach to microarray data analysis*. Norwell, MA: Springer, Kluwer.

Weijters, A., van der Aalst, W. M. P., & de Medeiros, A. K. A. (2006). Process mining with the heuristics miner algorithm. In *BETA working paper series WP 166*. Eindhoven University of Technology: Eindhoven.

Witten, I. H., Frank, E., & Hall, M. A. (2011). *Data mining: Practical machine learning tools and techniques*. San Francisco: Morgan Kaufmann Publishers Inc..

Xu, X., & Wang, X. (2005). An adaptive network intrusion detection method based on PCA and support vector machines. In X. Li, S. Wang, Z. Y. Dong (Eds.), *Advanced data mining and applications, first international conference (ADMA 2005)*, July 22–24, 2005. Wuhan, China: Proceedings. *Lecture notes in computer science* (Vol. 3584, pp. 696–703). Springer.

Ying, C. L. & Jin, A. T. B. 2007. Probabilistic random projections and speaker verification. In S. -W. Lee & S. Z. Li (Eds.), *Advances in biometrics, International conference (ICB 2007)*. August 27–29 2007. Seoul, Korea: Proceedings. *Lecture notes in computer science* (Vol. 4642, pp. 445–454). Springer.

Zhang, Z., Jiang, M., & Ye, N. (2010). Effective multiplicative updates for non-negative discriminative learning in multimodal dimensionality reduction. *Artificial Intelligence Review, 34*(3), 235–260.

Zho, Y., & Karypis, G. (2005). Hierarchical clustering algorithms for document datasets. *Data Mining and Knowledge Discovery, 10*(2), 141–168.