

# A Comparative Study of Dimensionality Reduction Techniques to Enhance Trace Clustering Performances



Hanna Yang

Technology Management/Information System/Entrepreneurship Program  
Graduate School of UNIST

2012

A Comparative Study of  
Dimensionality Reduction Techniques  
to Enhance Trace Clustering Performances

Hanna Yang

Technology Management/Information System/Entrepreneurship Program  
Graduate School of UNIST

# A Comparative Study of Dimensionality Reduction Techniques to Enhance Trace Clustering Performances

A thesis  
submitted to the Graduate School of UNIST  
in partial fulfillment of the  
requirements for the degree of  
Master of Science

Hanna Yang

07. 23. 2012

Approved by



---

Major Advisor

Minseok Song

# A Comparative Study of Dimensionality Reduction Techniques to Enhance Trace Clustering Performances

Hanna Yang

This certifies that the thesis of Hanna Yang is approved.

07. 23. 2012

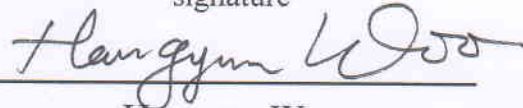
signature



---

Thesis supervisor: Minseok Song

signature



---

Hangyun Woo

signature



---

Duck Young Kim

## **Abstract**

Process mining aims at extracting useful information from event logs. Recently, in order to improve processes, several organizations such as high-tech companies, hospitals, and municipalities utilize process mining techniques. Real-life process logs from such organizations are usually very large and complicated, since the process logs in general contain numerous activities which are executed by many employees. Furthermore, lots of real-life process logs generate spaghetti-like process models due to the complexity of processes. Traditional process mining techniques have problems with discovering and analyzing real-life process logs which come from less structured processes. To overcome the weaknesses of traditional process mining techniques, a trace clustering has been developed. The trace clustering splits an event log into several subsets, and each subset contains homogenous cases. Even though the trace clustering is useful to handle complex process logs, it is time-consuming and computationally expensive due to a large number of features generated from complex logs.

In this thesis, we applied dimensionality reduction (preprocessing) techniques to the trace clustering in order to reduce the number of features. To validate our approach, we conducted experiments to discover relationships between dimensionality reduction techniques and clustering algorithms, and we performed a case study which involves patient treatment processes of a hospital. Among many dimensionality reduction techniques, we used three techniques namely singular value decomposition (SVD), random projection, and principal components analysis (PCA).

The result shows that the trace clustering with dimensionality reduction techniques produce higher average fitness values. Furthermore, processing time of trace clustering is effectively reduced with dimensionality reduction techniques. Moreover, we measured similarity between clustering results to observe the degree of changes in clustering results while applying dimensionality reduction techniques. The similarity is resulted differently according to used clustering algorithm.



## Contents

I. Introduction .....	1
II. Related Work .....	4
2.1 Process Mining.....	4
2.2 Trace Clustering.....	4
2.3 Dimensionality Reduction Techniques .....	5
III. Trace Clustering and Dimensionality Reduction Techniques .....	6
3.1 Trace Clustering.....	6
3.1.1 Trace Profiles.....	7
3.1.2 Clustering Techniques .....	8
3.1.3 Distance Measures .....	10
3.2 Dimensionality Reduction Techniques .....	11
IV. Research Framework: Optimal Combinations of Clustering Algorithms and Dimensionality Reduction Techniques .....	14
4.1 Experiment Procedures .....	16
4.2 Experiment Setups .....	16
4.3 Running Data .....	16
4.4 Evaluation Criteria .....	18
V. Computational Results and Discussion .....	21
5.1 Average Fitness.....	21
5.2 Processing Time .....	27
5.3 Similarity.....	30
VI. Conclusion.....	36

## List of Figures

Figure 1. An example of process model outcomes of the trace clustering .....	2
Figure 2. Process of the trace clustering.....	6
Figure 3. The example of the trace profiles .....	7
Figure 4. K-means clustering (K=3) .....	8
Figure 5. An example of dendrogram .....	9
Figure 6. An example of SOM result in ProM.....	9
Figure 7. The proposed trace clustering process by integrating clustering algorithms with dimensionality reduction techniques .....	14
Figure 8. Design of experiments.....	15
Figure 9. Enhanced event log filter in ProM.....	17
Figure 10. Process models of running data .....	17
Figure 11. An example of similarity calculation processes.....	20
Figure 12. The graphs of average fitness results (K-means clustering) .....	23
Figure 13. The graphs of average fitness results (AHC) .....	25
Figure 14. The graph of average fitness results (SOM) .....	26
Figure 15. The graphs of processing time results (K-means clustering).....	28
Figure 16. The graph of processing time results (AHC) .....	29
Figure 17. The graph of processing time results (SOM).....	30
Figure 18. The graphs of similarity results (K-means clustering) .....	32
Figure 19. The graphs of similarity results (AHC) .....	34
Figure 20. The graph of similarity results (SOM).....	35



## List of Tables

Table 1. Terms for distance measure .....	10
Table 2. The resulting logs of filtering.....	17
Table 3. Average fitness results (K-means clustering).....	22
Table 4. Average fitness results (AHC) .....	24
Table 5. Average fitness results (SOM) .....	26
Table 6. The best applicable dimensionality reduction techniques in terms of average fitness .....	26
Table 7. Processing time results (K-means clustering).....	27
Table 8. Processing time results (AHC) .....	29
Table 9. Processing time results (SOM) .....	29
Table 10. The best applicable dimensionality reduction techniques in terms of processing time.....	30
Table 11. Similarity results (K-means clustering).....	31
Table 12. Similarity results (AHC) .....	33
Table 13. Similarity results (SOM) .....	35
Table 14. The dimensionality reduction techniques having the highest similarity value .....	35

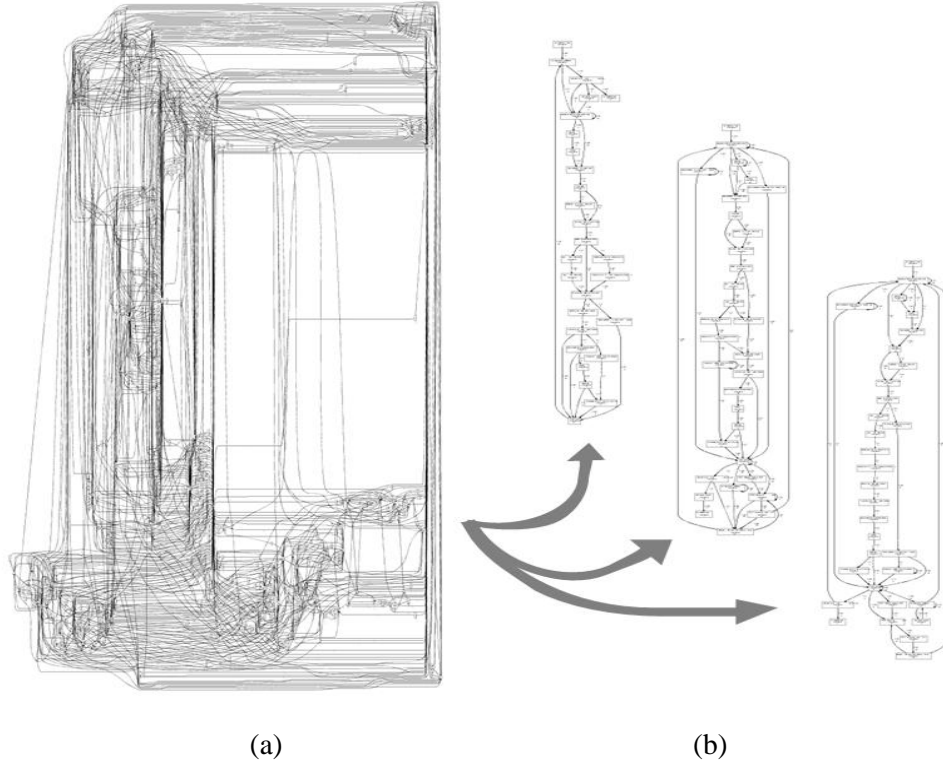
# I. Introduction

In order to realize competitive operational processes, organizations try to manage their processes more efficient. To achieve this goal, they need effective methods to analyze process execution results. Process mining is a technique for extracting useful information from process executions by analyzing event logs (van der Aalst et al., 2007, van der Aalst et al., 2004, Günther and van der Aalst, 2007). Through process mining, users can obtain business performance metrics, process models, organizational models, organizational relations, performance characteristics, and etc. (van der Aalst et al., 2007, Song and van der Aalst, 2008, Maruster and Beest, 2009, Günther and van der Aalst, 2007). Recently, several organizations such as high-tech companies, hospitals, and municipalities utilize process mining techniques to improve their processes (Song et al., 2008, Mans et al., 2008, Reijers et al., 2009, Lemos et al., 2011, Rozinat et al., 2009, van der Aalst et al., 2007).

Process mining techniques require less time and cost to analyze processes in comparison to the existing process analysis techniques such as business process reengineering (BPR), and six sigma. For example, in a BPR project, business process analysts gather process information by observing daily tasks and interviewing employees. It requires lots of time to collect process information and analyze business processes. However, process mining techniques require less time to collect process information since they use already collected process logs. Moreover, process mining techniques are more accurate than the existing process analysis techniques, since it helps analysts avoid possible personal biases during process analyses.

Traditional process mining techniques produce valuable information in various perspectives when they applied to well-structured processes which generate lasagna-process model (Jagadeesh Chandra Bose and van der Aalst, 2009, Günther and van der Aalst, 2007). However, lots of real-life business processes are unstructured processes which generate spaghetti-like process models. Real-life process logs are usually huge and complicated, since the process logs contain numerous activities which are executed by many employees. An example of a spaghetti-like process model is illustrated in Figure 1(a). The diversity of processes, i.e. each case has different kinds of activities as well as different sequences of activities, is a cause of spaghetti-like process model.

As illustrated in Figure 1(a), by observing the spaghetti-like process model, it is hard to discover useful information or conspicuous characteristics of process. In this case, we can use trace clustering to classify cases into homogeneous subsets (clusters) according to their log traces. Since cases in the same subset (cluster) have similar traces to each other, the process models of each cluster (Figure 1(b)) are much simpler than the process model out of a whole event log. Furthermore, it is much easier to extract useful information and find out problematic activities or employees from the process models of each cluster than the process model out of a whole event log.



**Figure 1:** An example of process model outcomes of the trace clustering

Despite the importance of trace clustering techniques, the trace clustering is time-consuming as well as computationally expensive due to too many features that most real-life business process logs contain. Furthermore, many features in the business process logs might have side effects on the trace clustering procedures, since they are trivial to be considered as features. Using all features from a process log, process mining results of each cluster can be inaccurate and useless due to the inaccurate trace clustering. In this thesis, we apply dimensionality reduction (preprocessing) techniques to the trace clustering in order to enhance trace clustering performances by reducing the number of features. Among many dimensionality reduction techniques, we used singular value decomposition (SVD), random projection, and principal components analysis (PCA).

We conducted experiments to discover relationships between dimensionality reduction techniques and clustering algorithms, and we used three evaluation criteria which are average fitness, processing time, and similarity. To validate our approach, we used a case study which involved patient treatment processes of a hospital. By applying the dimensionality reduction techniques to the trace clustering, average fitness value was improved. Also, processing time of trace clustering was effectively reduced with dimensionality reduction techniques. Similarity values, which are measured for the purpose of observing the degree of change in clustering results while applying the dimensionality reduction techniques, are resulted differently according to used clustering algorithm.

Business process analysts who employ the trace clustering might consider the results of this thesis for reference, when they need to reduce vector space of their logs.

The thesis is organized as follows. Related works are discussed in Section 2, Section 3 introduces trace clustering and dimensionality reduction techniques used in the thesis. Section 4 describes our research framework which includes experiment procedure, experiment setups, information of running data, and evaluation criteria. Section 5 presents results and Section 6 concludes the thesis.

## **II. 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, van der Aalst et al., 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 a time stamp (“when the event is executed”) of the event (van der Aalst and de Medeiros, 2005). Recently, process mining techniques are 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 organization such as public institutions (van der Aalst et al., 2007), manufacturers (Rozinat et al., 2009), telecom companies (Goedertier et al., 2011), and healthcare institutions (Mans et al., 2008), also it can be applied for internal fraud mitigation of organizations (Jans et al., 2011).

There exist three conceptual classes of process mining techniques which are discovery, conformance, and extension (Rozinat and van der Aalst, 2008). The concept of discovery aims at the creating a process models automatically from an event log (Jans et al., 2011, Rozinat and van der Aalst, 2008, Tsai et al., 2010). In general, it is a hard 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 process logs eg. the alpha algorithm (de Medeiros et al., 2003, van der Aalst et al., 2004), the heuristic miner (Weijters et al., 2006), the fuzzy miner (Günther and van der Aalst, 2007), and the genetic miner (de Medeiros and Weijters, 2005). The concept of conformance is about checking whether an existing process model matches a corresponding log, and measures for conformance checking such as fitness and appropriateness have been developed (Song et al., 2008, Rozinat and van der Aalst, 2008, Jagadeesh Chandra Bose and van der Aalst, 2009, Tsai et al., 2010). The concept of extension aims at the projecting information acquired from the event log onto the process model (Rozinat and van der Aalst, 2008, Maruster and Beest, 2009).

### **2.2 Trace Clustering**

Trace clustering has been discussed in many researches, because of the significance of the trace clustering to process mining. Greco et al. (Greco et al., 2006) used the trace clustering to classify cases of the event logs and facilitate the process of discovering expressive process models. In (Greco et al., 2006), the vector space model over the activities and transitions are used to find out proper

clusters. On the other hand, Song et al. proposed an approach to create profiles of the event log with control-flow perspective, organization perspective, and data perspective. The items included in the profiles are used as features which are the criteria of clustering algorithms. Therefore, Song et al. derives clusters based on not only activities and transitions, but also originators, data, performance, etc. as the feature vector (Song et al., 2008). Moreover, Jagadeesh Chandra Bose and van der Aalst studied the trace clustering which is based on a generic edit distance (Jagadeesh Chandra Bose and van der Aalst, 2009). To handle the sensitivity of the cost function when they used the generic edit distance framework, they proposed a method which automatically calculates the edit operations cost. Nevertheless, the trace clustering still has problems with the pitfalls highlighted as in (Jagadeesh Chandra Bose and van der Aalst, 2009). Overall, all clustering techniques are important methods in data mining field (Jain and Dubes, 1988). However, the clustering technique applied to the trace clustering in (Song et al., 2008) as well as this thesis are K-means clustering, agglomerative hierarchical clustering and self-organizing map, and they are the popular clustering algorithms in the data mining field.

### **2.3 Dimensionality Reduction Techniques**

A dimensionality of the data means that the number of attributes which describe every record in data. In data mining field, dimensionality reduction is an important problem since we are confronted with the problem of processing the high-dimensional data (Bartl et al., 2011, Zhao Zhang, 2010). Principal component analysis (PCA) and factor analysis (FA) are widely using dimensionality reduction techniques (Megalooikonomou et al., 2008, Bartl et al., 2011, Tan et al., 2006, Xu and Wang, 2005), and they are studied in many researches for a long periods. 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 attributes to quantitative attributes (Bartl et al., 2011). Multidimensional scaling (MDS) is a generalized technique of FA. MDS can be used to reduce dimensionality when the matrix is about the relationships between attributes or objects (Cil, 2012, Bécavin et al., 2011). Moreover, many dimensionality reduction techniques such as random projection (Bingham and Mannila, 2001, Johnson and Lindenstrauss, 1984, Achlioptas, 2003), singular value decomposition (Golub and Reinsch, 1970, Ma et al., 2001, Gong and Liu, 2000), and fisher discriminant analysis (Zhao Zhang, 2010) are developed and applied in many researches.

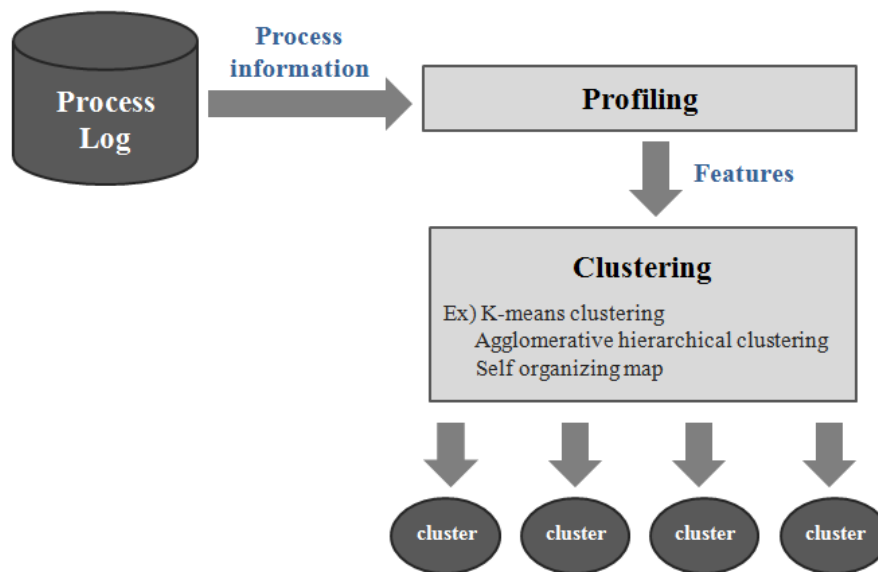
### III. Trace Clustering and Dimensionality Reduction Techniques

#### 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 cluster are much simpler than the process model out of a whole event log. Besides, by applying various process mining techniques to each cluster separately, we can extract useful information more easily because of the simplicity of the logs from each cluster.

The process of the trace clustering (Figure 2) is divided in two parts, one is profiling and another is 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, the clustering algorithms are used to classify cases of the log, and the clustering algorithms require a vector space to measure 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 the trace profile are used as criteria of the clustering algorithm in second phase.

In this thesis, we used two trace profiles, which are an activity profile and a transition profile, 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 that are used in this thesis.

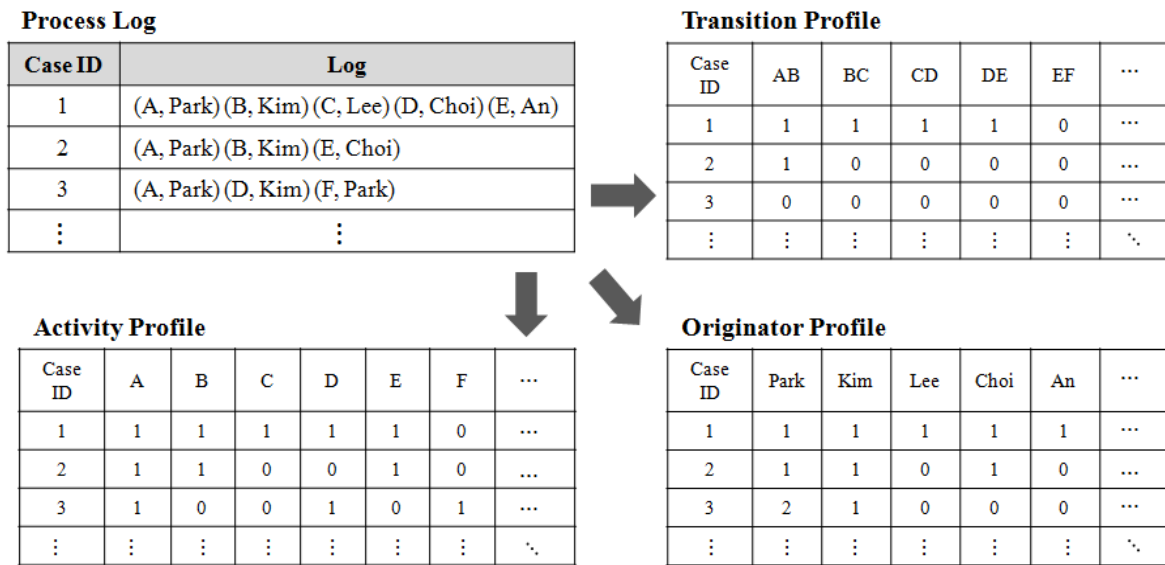


**Figure 2:** Process of the trace clustering

### 3.1.1 Trace Profiles

All clustering algorithms require criteria for classifying dataset. 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 the cases from a particular perspective, and every item in the trace profile can be used as a criterion for classifying cases in the clustering phase. Also, all values in the trace profile are expressed in numerical value.

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



**Figure 3:** The example of the trace profiles

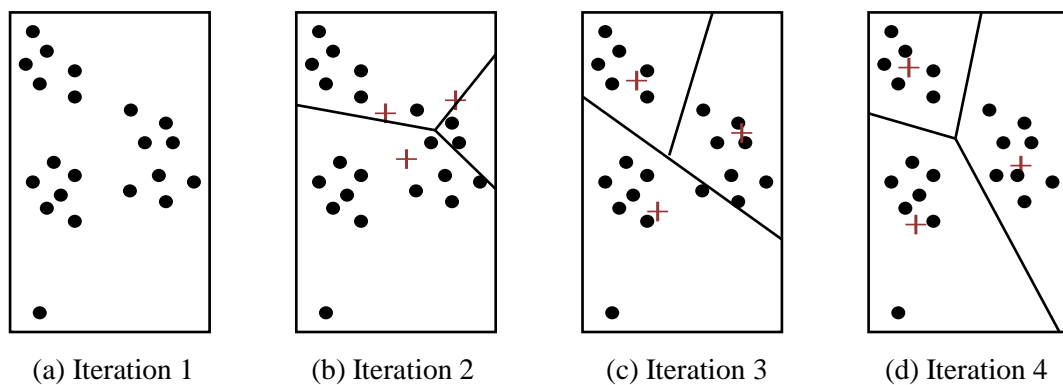


### 3.1.2 Clustering Techniques

#### *K-means Clustering*

K-means clustering algorithm is a frequently used partitioning method in practice (Song et al., 2008). By employing K-means clustering, we can obtain K clusters from a process log. Figure 4 shows that the example of K-means clustering process when K is 3. Each point in iteration 1 indicates each data. From iteration 2 to 4, the points included in different cluster are divided black lines to make them easy to figure out. First we need to select K initial centroid (center) points as illustrated in iteration 2 of figure 4, and make clusters by assigning each point to the closest centroid. Then, the centroid in each cluster moves to the mean distance point of the cluster that the centroid belongs to. Second and third steps are repeated until the centroids do not move (Tan et al., 2006). Initial centroids are randomly located and close to each other, but they move to the center of the each group of cases as the algorithm repeated. At iteration 4, the clustering is completed the way that minimize total distances between each case in the same cluster and maximize distances between the clusters.

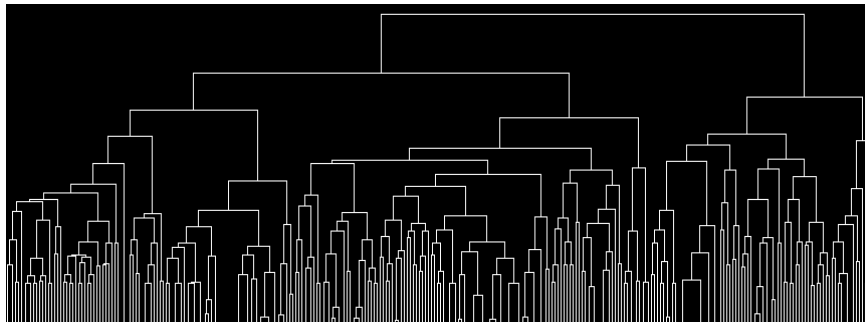
Even though multiple iterations are required to run the data, K-means clustering algorithm is very efficient algorithm in comparison to other clustering algorithms which are developed in the data mining field (Pelleg and Moore, 2000). Therefore, K-means clustering is still important subject of researches even it is developed and studied since 1967 (MacQueen, 1967). Many variations of K-means clustering, which are X-means clustering (Pelleg and Moore, 2000), K-harmonic means clustering, and other clustering algorithms have been constructed and studied to obtain better clustering results.



**Figure 4:** K-means clustering (K=3)

### *Agglomerative Hierarchical Clustering*

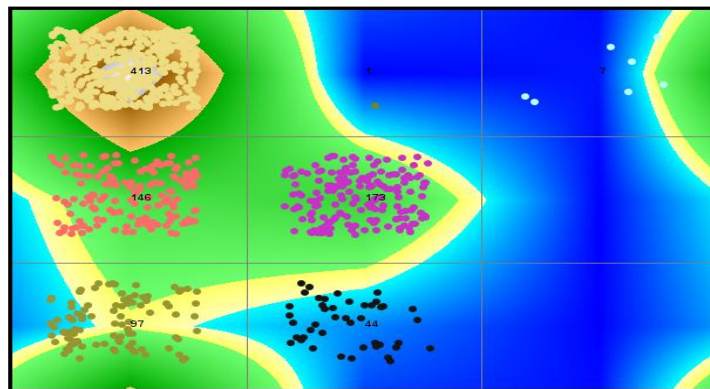
Agglomerative hierarchical clustering (AHC) is considered as the one of the important clustering technique in data mining field, since it has been studied relatively long time compared to other many kinds of clustering techniques (Tan et al., 2006). AHC algorithm starts with considering each point as a single cluster. Then clusters are merged according to distances between each cluster, and the same process is repeated until the number of cluster reaches to one (Zho and Karypis, 2005). AHC algorithm runs only once and creates a dendrogram which is a tree like diagram. Figure 5 shows an example of dendrogram, and the height of each node indicates proportional intergroup dissimilarity between two daughters of the cluster (Witten et al., 2011).



**Figure 5:** An example of dendrogram

### *Self Organizing Map*

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 process logs (Sarwar et al., 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. Figure 6 shows example of SOM result where each dot denotes each case and the cases belong to the same cluster expressed in the same color.



**Figure 6:** An example of SOM result in ProM

### 3.1.3 Distance Measures

To classify cases into clusters, the clustering algorithm needs a method to calculate the dissimilarities between any two cases. The cases can be projected in vector space based on the data in profiles, and measured distance between specific two cases in the vector space is the dissimilarity of those two cases. The methods to calculate distances between any two cases of the log are called ‘distance measures’. There are many kinds of distance measures such as hamming distance, jaccard index, and correlation coefficient (Song et al., 2008), and they are usually stemmed from data mining field. In this thesis, we used Euclidean distance to measure the dissimilarities between any two cases of the log.

#### *Euclidean Distance*

Through the profiles which are generated in the first phase of the trace clustering, we can project the cases of the log to an  $n$ -dimensional vector space. The  $n$  means the number of the features extracted from the process log to be used as criteria when we apply the clustering algorithm for classifying the cases of the process log. Terms that we need to understand for using and expressing the distance measure are explained in Table 1 (Song et al., 2008).

**Table 1:** Terms for distance measure

Term	Explanation
$c_j$	Corresponds to the vector $\langle i_{j1}, i_{j2}, \dots, i_{jn} \rangle$
$i_{jk}$	The number of appearance of item $k$ in the case $j$
$k$	$k$ th item (feature or activity)
$j$	$j$ th case
$n$	The number of features extracted from process log to be criteria of clustering algorithm

The Euclidean distance is used for computing a similarity between two vectors; it can calculate the similarity efficiently between two vectors regardless of the dimension of the vector space (Jeong et al., 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. The Euclidean distance is defined as follow (Duda et al., 2000) :

$$\text{Euclidean distance } (c_j, c_k) = \sqrt{\sum_{l=1}^n \| i_{jl} - i_{kl} \|^2} \quad (1)$$

### 3.2 Dimensionality Reduction Techniques

Dimensionality reduction (preprocessing) techniques are studied in data mining field for many years to classify and cluster databases. In the data mining field, as the methods of collecting data are developing, the features that are used to cluster the data become much bigger while many of them are irrelevant and redundant. Therefore, the dimensionality reduction techniques are proposed to deal with these challenging tasks involving many irrelevant and redundant features and often comparably few training examples. Among many preprocessing techniques, we use singular value decomposition, random projection and principal components analysis in this thesis.

#### *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:

$$M = U\Sigma V^*$$

In the equation,  $M$  is an  $m \times n$  matrix which consists of real numbers and complex numbers, and the entries of  $M$  are component of dataset. In this thesis, each column represents each case and each row represents each feature created by profiling. According to SVD equation,  $M$  is decomposed to three matrices which are  $U$ ,  $\Sigma$ ,  $V^*$ . The matrix  $U$  denotes an  $m \times m$  orthogonal transformation matrix, the matrix  $\Sigma = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_n)$  is an  $m \times n$  diagonal matrix, and an  $n \times n$  unitary matrix  $V^*$  denotes the conjugate transpose of the matrix  $V$  (Wall et al., 2003). The diagonal entries ( $\sigma_i$ ) of the matrix  $\Sigma$  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 satisfies (Gong and Liu, 2000)

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_r \geq \sigma_{r+1} = \dots = \sigma_n = 0$$

In this thesis, by selecting  $k$ -largest singular values, we can project the data to  $k$  dimension space. The  $\sigma_i$ s whose  $i$  is larger than  $k$  are set to 0, and then calculate reduced matrix  $M_k$ . Then, the data in the matrix  $M_k$  are projected to  $k$  dimension space. SVD is an excellent and powerful technique in many fields. For example, it can be implemented in signal modeling, system identification, image reconstruction, realization, reliable computations, and etc (Ma et al., 2001). We can use SVD to attain the immunity from noise effects. Also SVD disuses small singular values to solve ill-conditioned linear equations (Golub and Reinsch, 1970). 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

the 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). Moreover, SVD has been used in the fields such as text retrieval (Nicholas and Dahlberg, 1998), video summarization (Gong and Liu, 2000), and hand gesture recognition (Liu and Kavakli, 2010).

### *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:

$$X_{k \times N}^{RP} = R_{k \times d} X_{d \times N}$$

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 and 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 and Lindenstrauss, 1984). Moreover, it is computationally very efficient and has very strong probabilistic foundations (Achlioptas, 2003). Random projection has been applied to various data such as text data, image data (Bingham and Mannila, 2001), and cancellable biometrics approaches in face recognition (Ying and Jin, 2007), in order to reduce the dimensionality of data.

### *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 et al., 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:

$$C v_i = \lambda_i v_i$$

The matrix  $C$  is covariance matrix of vectors of the original data  $X$ , and  $\lambda_i$ 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 correspond to the  $k$  largest eigenvalues need to be computed (Xu and Wang, 2005).

Let

$$E_k = [v_1, v_2, v_3, \dots, v_k] \text{ and } \Lambda = [\lambda_1, \lambda_2, \lambda_3, \dots, \lambda_k],$$

then we have

$$C E_k = E_k \Lambda$$

Then, finally we can obtain the equation

$$X^{PCA} = E_k^T X$$

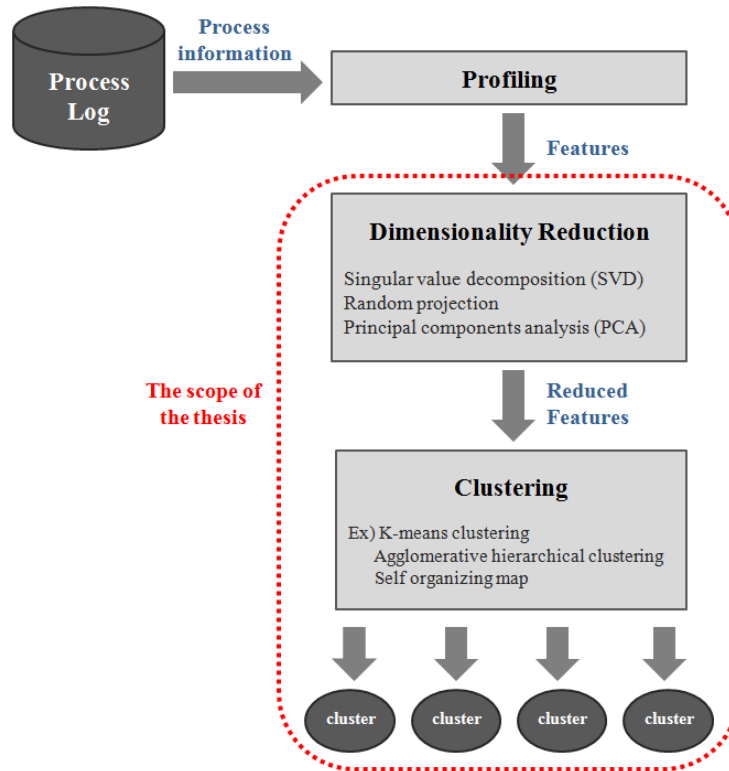
According to the equation, the number of the features of the original data matrix  $X$  is reduced by multiplying with a  $d \times k$  matrix  $E_k$  which has  $k$  eigenvectors corresponding to the  $k$  largest eigenvalues. The result matrix is  $X^{PCA}$  (Bingham and Mannila, 2001).

Moreover, PCA uses clustering to predict user preferences (Goldberg et al., 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 et al., 2004).

PCA is closely related to SVD. PCA aims to find out the basis which can express the original data more meaningful way. The goal of PCA is a change of basis, and a more general technique about the change of basis is SVD. As explained before, the eigenvectors of the matrix  $C$  are the principal components of the original data matrix  $X$ . Moreover, the columns of the matrix  $V$  in SVD can contain the eigenvectors of the matrix  $C$  in PCA, if we apply SVD to the matrix  $\frac{1}{\sqrt{n}}X^T$ . It can be interpreted as that the column space of the matrix  $\frac{1}{\sqrt{n}}X$  is covered by the matrix  $V$  (Shlens, 2005).

## IV. Research Framework: Optimal Combinations of Clustering Algorithms and Dimensionality Reduction Techniques

There are a large number of the features in the profiles of the process logs that we use to test our experiments, and using all features as criteria for the clustering algorithm is too computationally expensive. Furthermore, some of the features should not be used as criteria for the clustering algorithm. To overcome the challenges of the trace clustering, we applied dimensionality reduction techniques to the trace clustering as illustrated in Figure 7. By applying the dimensionality reduction techniques, we can provide reduced number of the features to the clustering algorithms as clustering criteria.

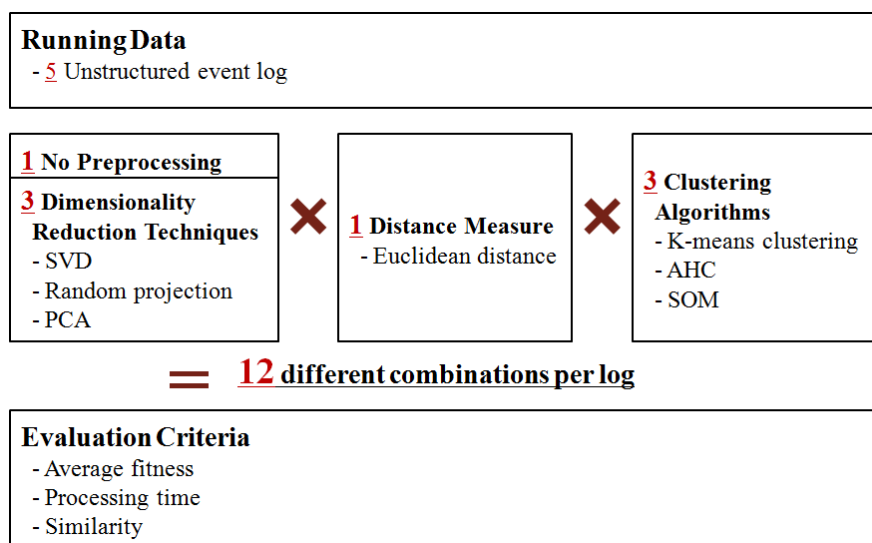


**Figure 7:** The proposed trace clustering process by integrating clustering algorithms with dimensionality reduction techniques

We aimed to discover relationships between the dimensionality reduction techniques and the clustering algorithms, and we used three evaluation criteria which are an average fitness, a processing time, and a similarity. The average fitness is an average of fitness values derived from clusters which are generated by the trace clustering. The processing time shows the time to produce trace clustering

results and it is required to be measured in order to show the efficiency of the dimensionality reduction techniques. The similarity is calculated as a rate of match between the clustering result when preprocessing is used and when it is not used. The rate of match was computed by comparing case ids that each cluster contains.

Our design of the experiments is presented in Figure 8. We used five real-life process logs for experiments. They are unstructured event logs, and they are basically same hospital logs but have different complexities of the log compositions. Details about the event logs are in section 4.3. Also, three dimensionality reduction techniques which are singular value decomposition (SVD), random projection, and principal components analysis (PCA) are used. Moreover, to estimate the influence of dimensionality reduction techniques to trace clustering results, we generated the trace clustering results without preprocessing. Among many clustering algorithms have been developed, we used three clustering algorithms which are K-means clustering, agglomerative hierarchical clustering (AHC), and self-organizing map (SOM). The cases can be projected in vector space based on the data in profiles, and distance between specific two cases in the vector space is interpreted as the dissimilarity of those two cases. The distance measure is a method to calculate distance between two cases in the vector space. In the thesis, among many distance measures such as hamming distance, jaccard index, and correlation coefficient, we used Euclidean distance as the distance measure of the experiments. As illustrated in Figure 8, 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 of 12 combinations. To compare results of 12 combinations, we used three evaluation criteria which are the average fitness, the processing time, and the similarity. Details about the evaluation criteria are in section 4.4.



**Figure 8:** Design of the experiments



## 4.1 Experiment Procedures

The process of the experiments is as follows. First, we implement the trace clustering to the experimental logs and achieve the 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 the trace clustering, we need reference trace clustering results which do not affected by any kind of variables. Second, we start implement the trace clustering with Euclidean distance, one of preprocessing techniques, and one of clustering algorithms. Since, there are three clustering algorithms and three preprocessing techniques that we use in the experiments; we can derive nine different trace clustering results per log. Totally we can get 12 different results per log including the control variable results. Last, we compare and evaluate outcomes. The comparison should be executed among the results that use the same clustering algorithm. In other words, results from 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 3.20GHz (4 CPUs) with 3072MB RAM and Windows 7 Enterprise K 32-bit operating System.

We use ProM 5.2 tool to test our experiments. ProM is an effective framework for performing process mining techniques which is able to analyze XES or MXML format process 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 (Process Mining Group, 2009).

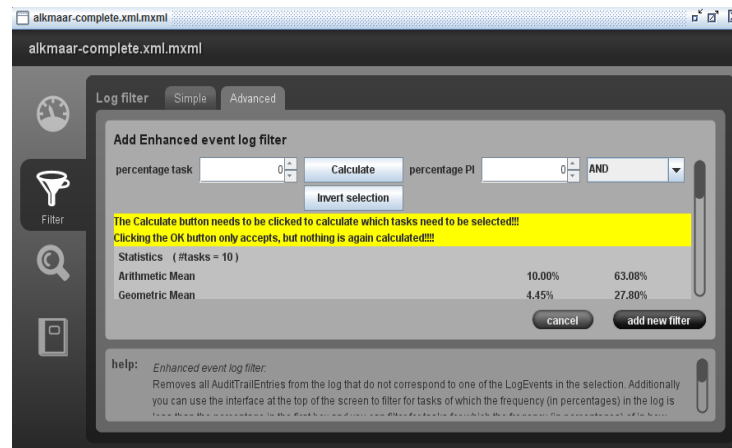
## 4.3 Running Data

We use extracted event log from the AMC hospital's databases to test our theory. 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, 1,143 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 provided from ProM in Figure 9. By using Enhanced event log filter, user can remove events which occurred less than particular rate in the entire log, and generate filtered event log separately from the original event log. Table 2 lists the resulting logs and the information of them. In Table 2, 0.3% filtered log means that the log does not contain the events appeared less than 0.3% in the entire log. Figure 10 shows two process models which are generated  $PL_1$  and  $PL_5$ , and it is easy to understand the difference of complexities between two logs by comparing two models. The process model generated based on  $PL_1$ , which is the simplest log, is in

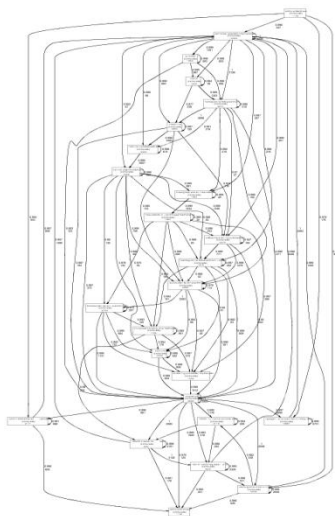
Figure 10(a). Also, the process model generated based on  $PL_5$ , which is unfiltered log, is in Figure 10(b). From unstructured process model as the model in Figure 10(b), it is hard to extract useful information because the model is too complex and containing too many activities and relations of activities.

**Table 2:** The resulting logs of filtering

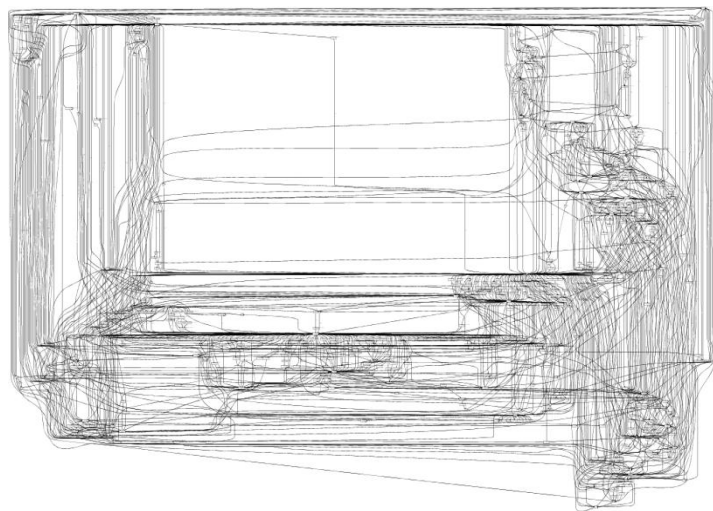
Log name	Filtering (%)	# of events per case			# of types of event
		min	average	max	
$PL_1$	1.0	3	18	25	25
$PL_2$	0.8	3	22	32	32
$PL_3$	0.5	3	28	48	49
$PL_4$	0.3	3	31	63	65
$PL_5$	0	1	33	113	624



**Figure 9:** Enhanced event log filter in ProM



(a) Process model of  $PL_1$



(b) Process model of  $PL_5$

**Figure 10:** Process models of running data

#### 4.4 Evaluation Criteria

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

##### *Average Fitness*

The first evaluation criterion is the average fitness. Fitness value explains how well an event log fits its process model. If the process model can regenerate traces of all cases in the log, we can say that the log fits the process model (Rozinat and van der Aalst, 2008). According to Rozinat and van der Aalst, 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 (fired) in the process model called Petri net, and the details about the token and Petri net are in (Rozinat and van der Aalst, 2008) and (de Medeiros et al., 2003). After counting tokens according to their conditions, we put those numbers in the fitness equation. The fitness equation is defined as follow:

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

In the equation, the number of cases is expressed as  $k$ ,  $m_i$  is the number of missing tokens. Also  $c_i$  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 is 1. In our experiments, we measured the fitness of each cluster and calculated the average of all fitness values, so we used 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.

##### *Processing Time*

The second evaluation criterion is the processing time. By comparing the processing time of the trace clustering with the dimensionality reduction techniques and the one without dimensionality reduction techniques, the effect of applying the preprocessing on the trace clustering can be explained. In our experiments, we measured the processing time of the trace clustering in seconds. 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.

### *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.

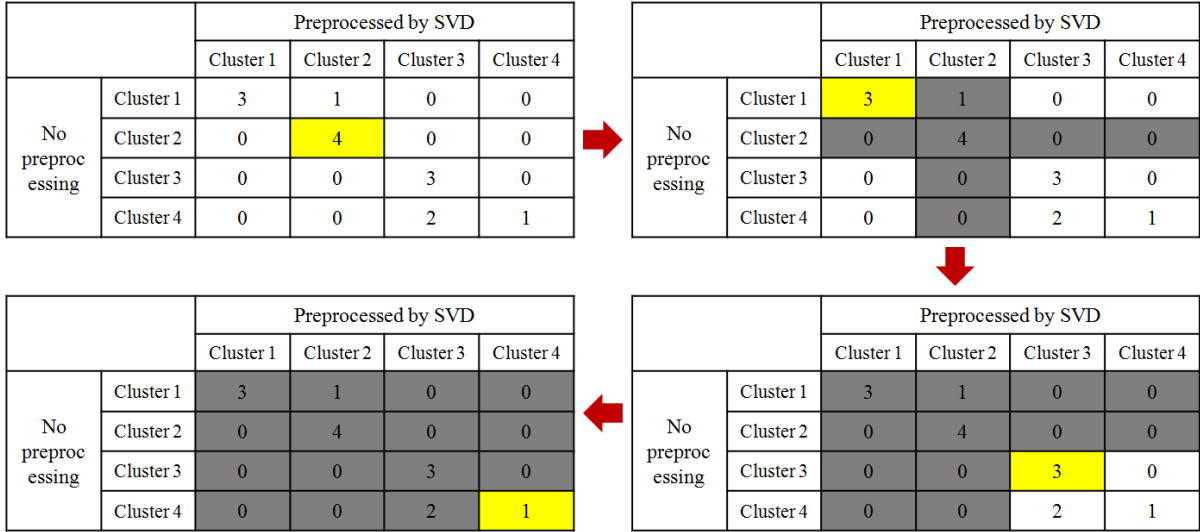
Figure 11 shows an example of the similarity calculation processes. In the example, we compared the trace clustering results without preprocessing and the trace clustering results preprocessed by SVD. First, we need to obtain the case ids of each cluster in both results as in Figure 11(a). Then, generate a similarity matrix as in Figure 11(b). Values in the blank of the similarity matrix mean the number of case ids that both clusters contain identically. 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 next highest value in the same row or column. The whole example processes are in Figure 11(c). Through the processes in Figure 11(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. Figure 11(d) shows the outcomes resulted from the process in Figure 11(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  $(3+4+3+1)/14 = 0.7857$ .

No preprocessing		Preprocessed by SVD	
	Case ID		Case ID
Cluster 1	1, 2, 3, 4	Cluster 1	1, 2, 3
Cluster 2	5, 6, 7, 8	Cluster 2	4, 5, 6, 7, 8
Cluster 3	9, 10, 11	Cluster 3	9, 10, 11, 12, 13
Cluster 4	12, 13, 14	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



(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)

**Figure 11:** An example of similarity calculation processes

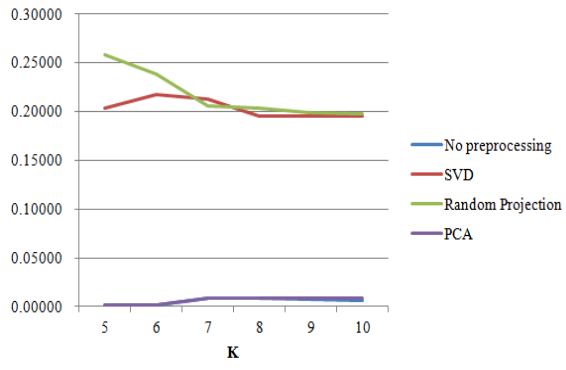
## V. Computational Results and Discussion

### 5.1 Average Fitness

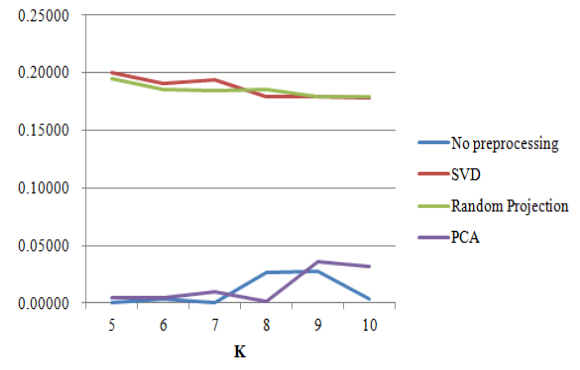
Although we used filtering to reduce the complexity of the logs, the average fitness values are very low due to the complexities of the logs. The average fitness results, when we use K-means clustering with different preprocessing techniques, are in Table 3. To do a comparative analysis of the average fitness values in Table 3, we draw graphs of the results as shown in Figure 12. The graphs show the average fitness values of each log when we use K-means clustering with different preprocessing techniques. The horizontal axis of the graph represents the K value, and the vertical axis of the graph represents the average fitness value. Therefore, we can conclude that when we implement the trace clustering to  $PL_I$ , the combination of random projection and K-means clustering is the best combination in terms of average fitness except when K is 7. The exception can be interpreted in terms of optimal K, but it is not the focus of this thesis. Moreover, we obtained the fact that the size and the complexity of the log can affect the results of the experiments.

**Table 3:** Average fitness results (K-means clustering)

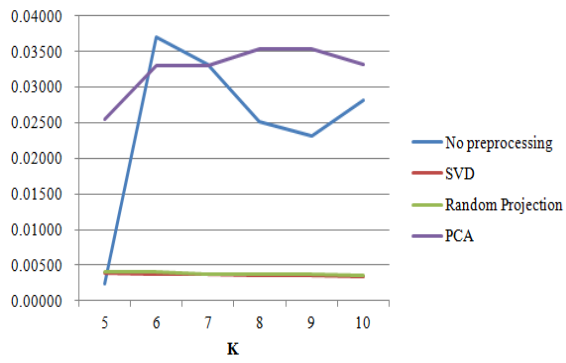
Log name	K	No preprocessing	SVD	Random projection	PCA
$PL_1$	5	0.00104	0.20326	0.25792	0.00205
	6	0.00120	0.21747	0.23837	0.00197
	7	0.00829	0.21229	0.20589	0.00834
	8	0.00824	0.19591	0.20397	0.00808
	9	0.00699	0.19564	0.19880	0.00826
	10	0.00640	0.19591	0.19755	0.00806
$PL_2$	5	0.00000	0.20030	0.19515	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



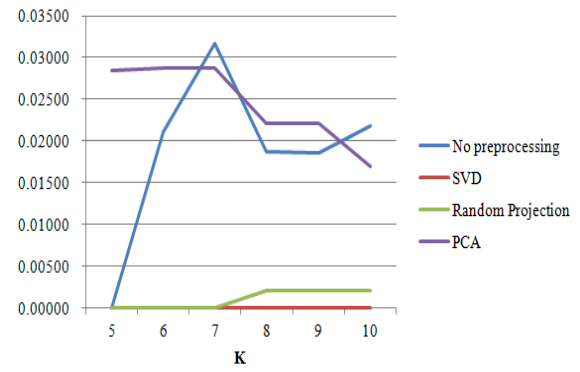
(a)  $PL_1$



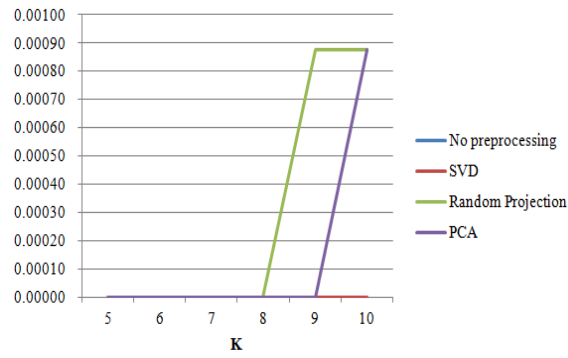
(b)  $PL_2$



(c)  $PL_3$



(d)  $PL_4$



(e)  $PL_5$

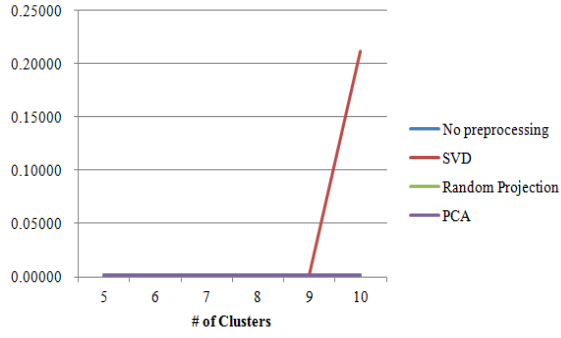
**Figure 12:** The graphs of average fitness results (K-means clustering)



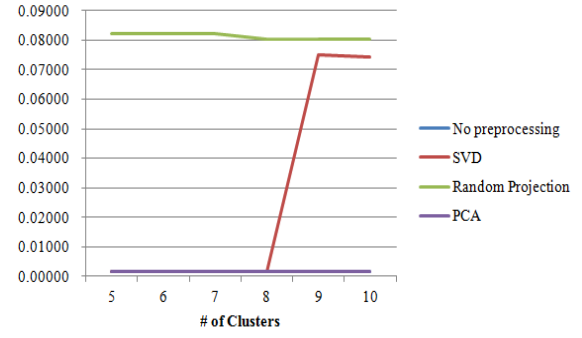
Average fitness results, when we use AHC with different preprocessing techniques, are listed in Table 4. The graphs in Figure 13 show the average fitness values of each log when we use AHC with different preprocessing techniques. The horizontal axis of the graph represents the number of clusters, and the vertical axis of the graph represents the average fitness value.

**Table 4:** Average fitness results (AHC)

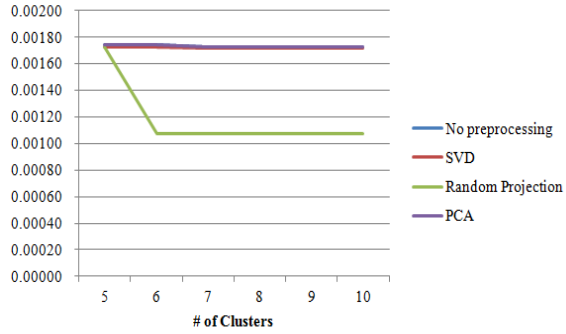
Log name	# of Clusters	No preprocessing	SVD	Random projection	PCA
$PL_1$	5	0.00172	0.00172	0.00169	0.00174
	6	0.00172	0.00171	0.00169	0.00174
	7	0.00172	0.00171	0.00163	0.00173
	8	0.00172	0.00171	0.00163	0.00173
	9	0.00172	0.00171	0.00163	0.00173
	10	0.00172	0.21143	0.00163	0.00173
$PL_2$	5	0.00173	0.00174	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



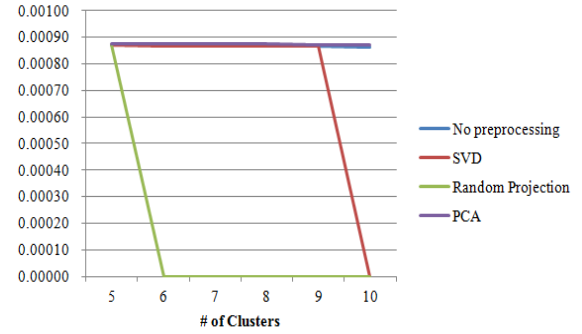
(a)  $PL_1$



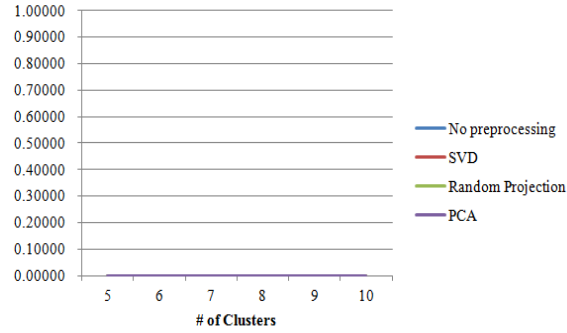
(b)  $PL_2$



(c)  $PL_3$



(d)  $PL_4$



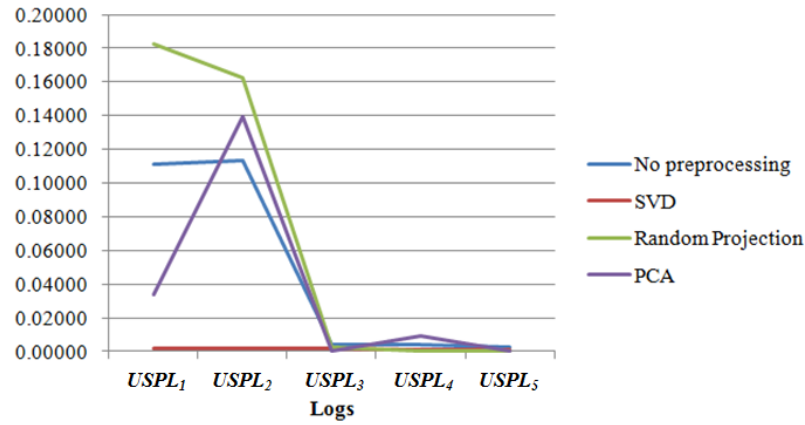
(e)  $\text{Log } PL_5$

**Figure 13:** The graphs of average fitness results (AHC)

The average fitness results, when we use SOM with different preprocessing techniques, are in Table 5. Since SOM does not require predetermined number of clusters, each log has four results. The graph in Figure 14 shows the average fitness values when we use SOM with different preprocessing techniques. The horizontal axis of the graph represents name of the log, and the vertical axis of the graph represents the average fitness value.

**Table 5:** 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

**Figure 14:** The graph of average fitness results (SOM)

The best dimensionality reduction techniques in terms of average fitness are organized in Table 6 by the clustering algorithm and the log name.

**Table 6:** The best applicable dimensionality reduction techniques in terms of average fitness

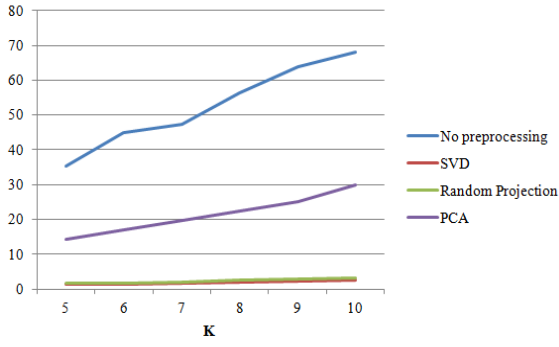
Log name	K-means clustering	AHC	SOM
$PL_1$	SVD Random projection	SVD	Random projection
$PL_2$	SVD Random projection	Random projection	Random projection
$PL_3$	PCA	No preprocessing SVD PCA	No preprocessing
$PL_4$	PCA	PCA	PCA
$PL_5$	Random projection	No preprocessing SVD Random projection PCA	No preprocessing

## 5.2 Processing Time

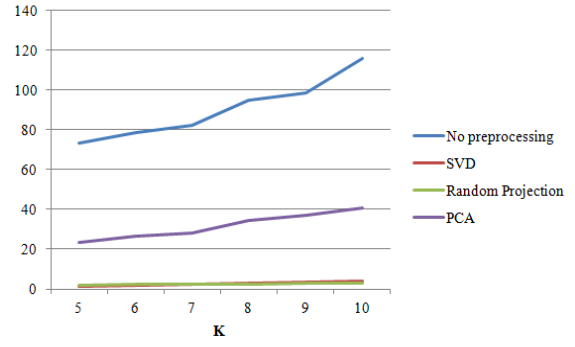
Table 7 lists the processing time results of the logs when we use K-means clustering with various preprocessing techniques. To do a comparative analysis of the processing time results in Table 7, we draw graphs of the results as shown in Figure 15. The graphs show the processing time of each log when 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).

**Table 7:** Processing time results (K-means clustering)

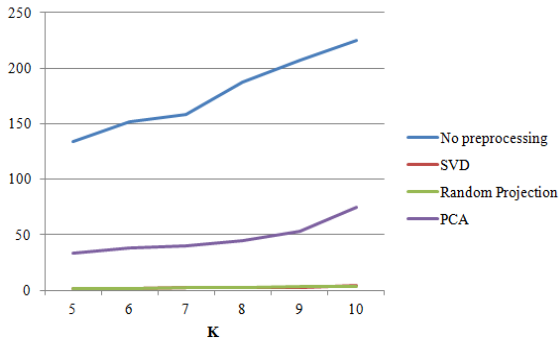
Log name	K	No preprocessing	SVD	Random projection	PCA
$PL_1$	5	35.3	1.5	1.7	14.3
	6	44.9	1.5	1.8	17.1
	7	47.2	1.8	1.9	19.7
	8	56.3	2.0	2.7	22.5
	9	64.0	2.4	3.0	25.0
	10	68.2	2.6	3.1	29.8
$PL_2$	5	73.4	1.4	2.0	23.5
	6	78.9	1.9	2.2	26.7
	7	82.1	2.5	2.3	28.4
	8	95.2	2.9	2.6	34.2
	9	98.7	3.4	2.7	37.2
	10	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



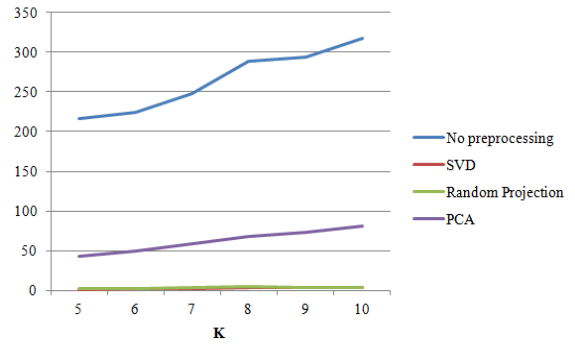
(a)  $PL_1$



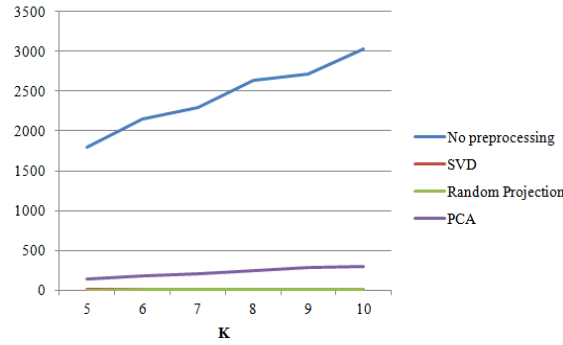
(b)  $PL_2$



(c)  $PL_3$



(d)  $PL_4$



(e)  $PL_5$

**Figure 15:** The graphs of processing time results (K-means clustering)

Table 8 lists the processing time results of the logs when we use AHC with various preprocessing techniques. There is only one processing time record for each log, when we use AHC as clustering algorithm of the trace clustering. Therefore we could acquire one graph as shown in Figure 16. The graph 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).

**Table 8:** 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

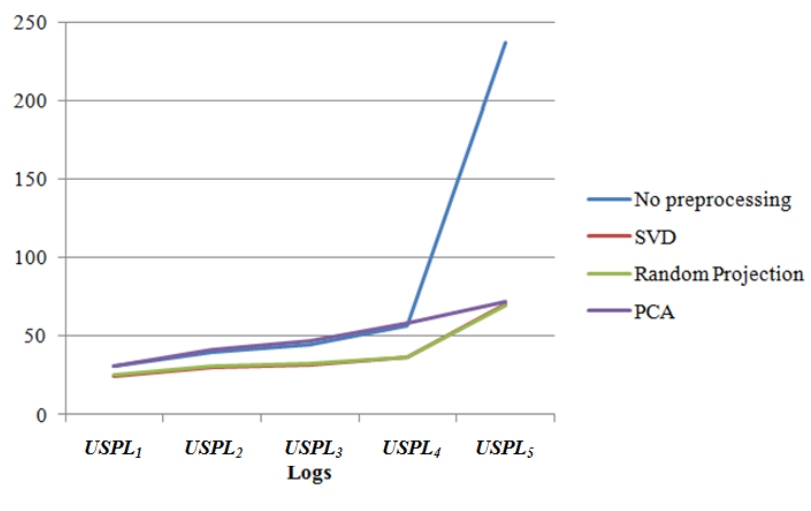
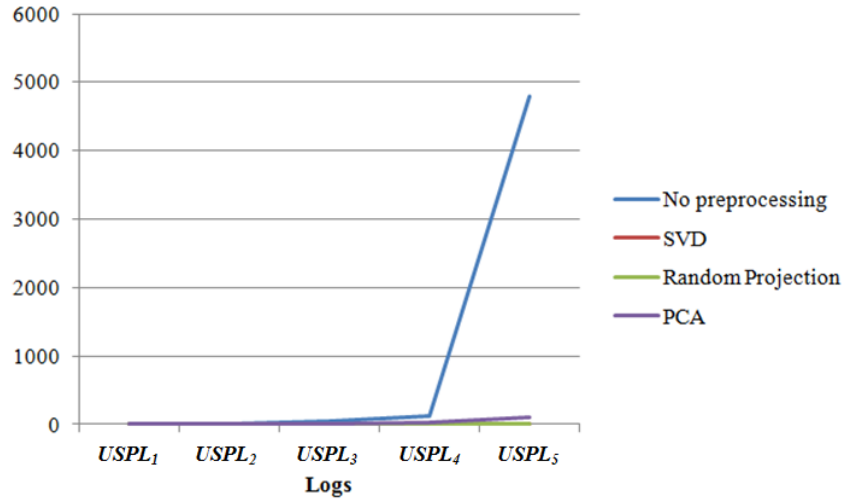
**Figure 16:** The graph of processing time results (AHC)

Table 9 lists the processing time of the logs when we use SOM with various preprocessing techniques. There is only one time record for each log, when we use SOM as clustering algorithm of the trace clustering. Therefore, we could obtain one graph as appeared in Figure 17. The graph in Figure 17 shows the processing time of each log when we use SOM 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 processing time to cluster cases (in seconds).

**Table 9:** 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



**Figure 17:** The graph of processing time results (SOM)

Table 10 lists the best dimensionality reduction techniques in terms of the processing time, and the outcomes are organized by the clustering algorithm and the log name. According to the results, when we use the trace clustering, it is better to apply SVD or random projection to decrease the clustering time significantly regardless the clustering algorithm that we use.

**Table 10:** 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
$PL_2$			
$PL_3$			
$PL_4$			
$PL_5$			

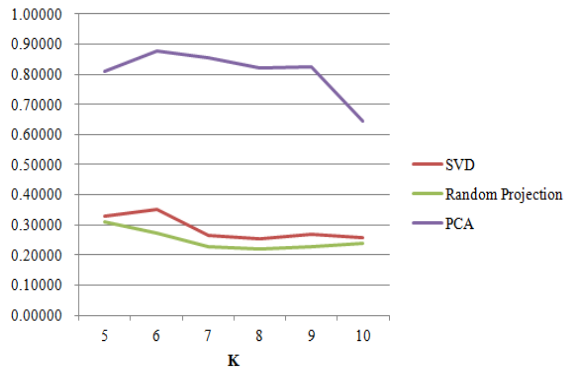
### 5.3 Similarity

We calculated a similarity by comparing one result and its relevant control variable result, so the column for ‘No preprocessing’ does not exist. The rates of match values, when we use K-means clustering with different preprocessing techniques, are calculated and listed in Table 11. To do a comparative analysis of the similarity values in Table 11, we draw the graphs of the results as shown in Figure 18. The graphs in Figure 18 show the similarity values of each log when we use K-means clustering while applying different preprocessing techniques. The horizontal axis of the each graph represents the K value, and the vertical axis of the each graph represents the rate of match to control variable result.

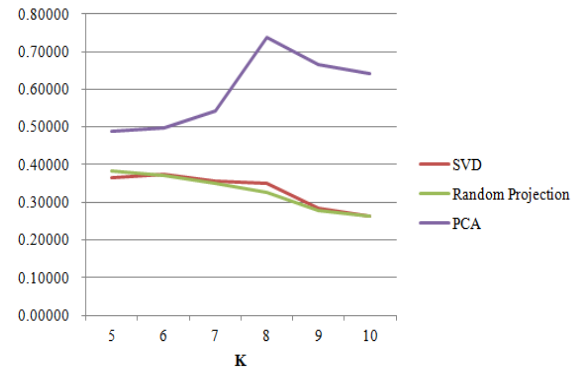
**Table 11:** Similarity results (K-means clustering)

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

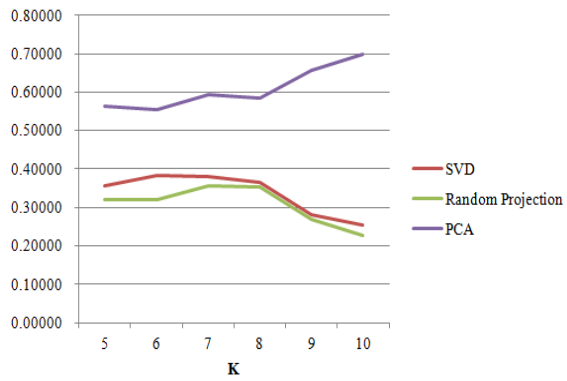




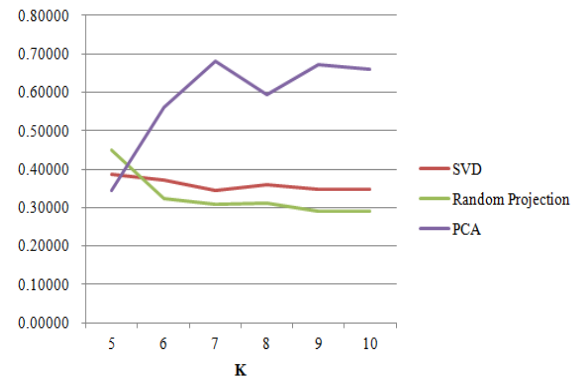
(a)  $PL_1$



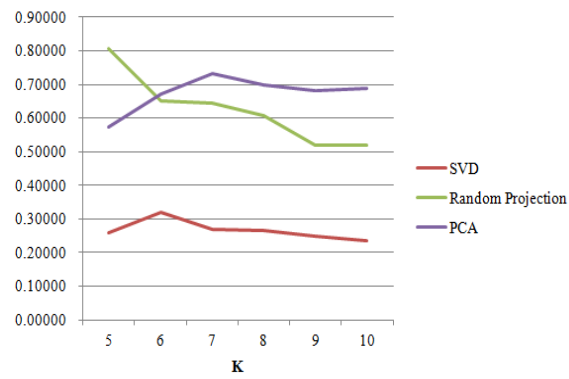
(b)  $PL_2$



(c)  $PL_3$



(d)  $PL_4$



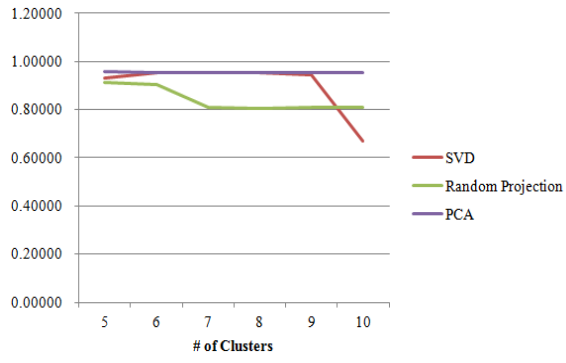
(e)  $PL_5$

**Figure 18:** The graphs of similarity results (K-means clustering)

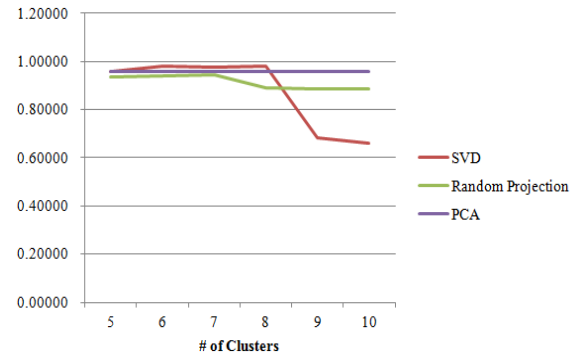
The rates of match values, when we use AHC with different preprocessing techniques, are calculated and listed in Table 12. Figure 19 shows the graphs of the similarity values of each log when we use AHC while applying different preprocessing techniques. The horizontal axis of the graph represents the number of clusters, and the vertical axis of the graph represents the rate of match to control variable result.

**Table 12:** Similarity results (AHC)

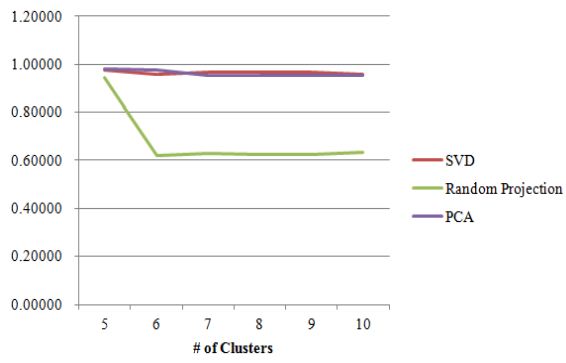
Log name	# of Clusters	SVD	Random projection	PCA
$PL_1$	5	0.93290	0.91380	0.95810
	6	0.95210	0.90540	0.95330
	7	0.95570	0.80840	0.95570
	8	0.95570	0.80600	0.95570
	9	0.94370	0.80840	0.95330
	10	0.67070	0.80960	0.95210
$PL_2$	5	0.95730	0.93480	0.95840
	6	0.97980	0.94160	0.95960
	7	0.97750	0.94270	0.95960
	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



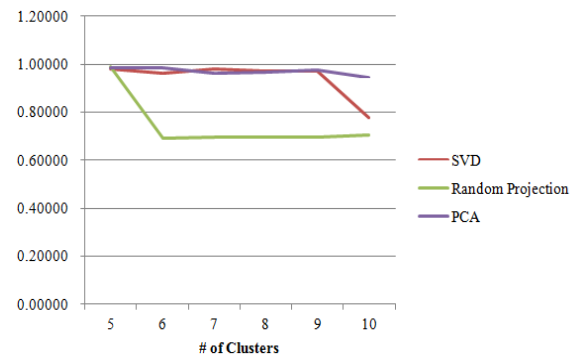
(a)  $PL_1$



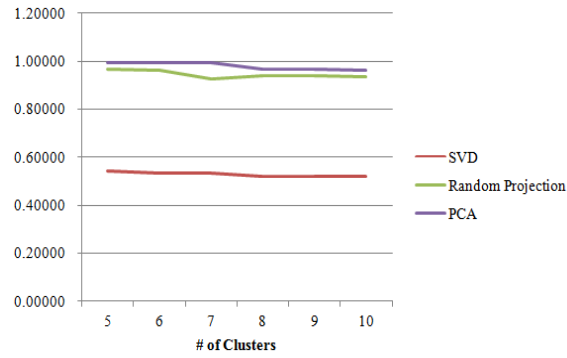
(b)  $PL_2$



(c)  $PL_3$



(d)  $PL_4$



(e)  $PL_5$

**Figure 19:** The graphs of similarity results (AHC)

The rates of match values, when we use SOM with different preprocessing techniques, are calculated and shown in Table 13. Figure 20 shows the graph of the similarity values of each log when we use SOM while applying different preprocessing techniques. The horizontal axis of the graph represents the number of clusters, and the vertical axis of the graph represents the rate of match to control variable result.

**Table 13:** 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

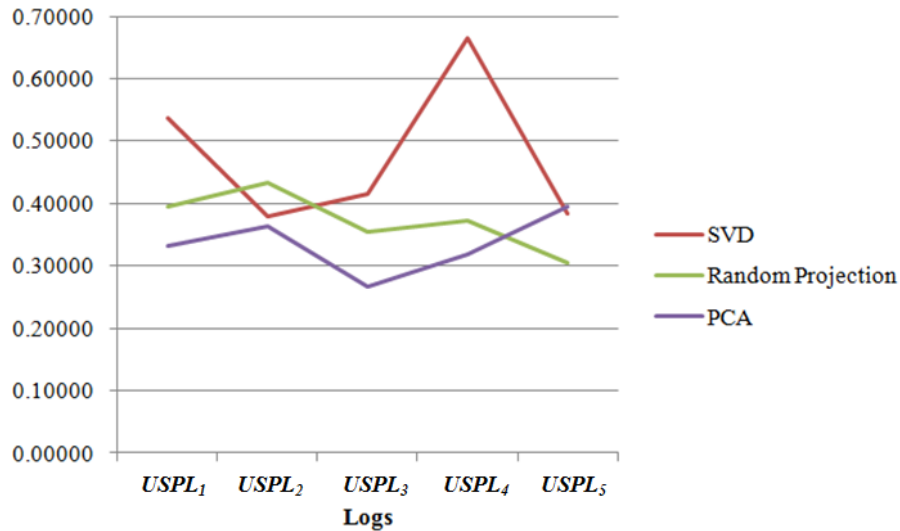
**Figure 20:** The graph of similarity results (SOM)

Table 14 shows the dimensionality reduction techniques which have the highest similarity values, the results are classified by the log name and the clustering algorithm that are used. According to the Table 14, the combination of K-means and PCA results the highest similarity value when it is applied to the trace clustering, and SVD and PCA are good dimensionality reduction techniques to be used with AHC. Also, the combination of SVD and SOM results the high similarity value when it is applied to the trace clustering.

**Table 14:** The dimensionality reduction techniques having the highest similarity value

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

## VI. Conclusion

In this thesis, we applied the preprocessing techniques to enhance the performances of the trace clustering which is used in the process mining analysis. We conducted the experiments to discover relationships between dimensionality reduction techniques and clustering algorithms. Also, we used a case study which involves patient treatment processes of a hospital to validate our approach.

We evaluated the results separately in terms of fitness, processing time, and similarity criteria. According to the results, 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 the dimensionality reduction techniques, we could enhance trace clustering performances. Similarity values are resulted differently according to used clustering algorithm.

The conclusions can be summarized as follow. First, the results about the best applicable dimensionality reduction techniques in terms of fitness could be various according to the complexity of the log and the used clustering algorithm. We could not find out any kind of trend from the average fitness results. 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 log or type of the clustering algorithm. Third, the dimensionality reduction techniques which results the highest similarity values are PCA for K-means clustering, SVD and PCA for AHC, SVD for SOM.

As for the future work, more research about the optimal applicable dimensionality reduction techniques to specific clustering algorithm of the trace clustering should be conducted regarding all three criteria (i.e. fitness, processing time and similarity) simultaneously. Furthermore, similar studies with other clustering algorithms and dimensionality reduction techniques are necessary. Moreover, similar studies with process logs of other industries are needed and recommended to prove the results of this thesis. Through further in-depth study, guidelines about the appropriate technique of dimensionality reduction for specific clustering algorithm of the trace clustering technique can be proposed. The proposed guideline will help business process analysts choose appropriate preprocessing techniques according to the particular nature of their business processes.

## Reference

- 1 Achlioptas, D. 2003. 'Database-friendly random projections: Johnson- Lindenstrauss with binary coins'. *Journal of Computer and System Sciences* 66(4), pp. 671–687.
- 2 Bécavin, C., Tchitchek, N., Mintsá-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), pp. 1413-1421.
- 3 Bartl, E., Rezanková, H. & Sobisek, L. 2011. 'Comparison of Classical Dimensionality Reduction Methods with Novel Approach Based on Formal Concept Analysis'. in Yao, J., Ramanna, S., Wang, G. & Suraj, Z. (eds), *Rough Sets and Knowledge Technology (RSKT 2011)*, October 9-12 2011, Banff, Canada. *Lecture Notes in Computer Science* 6954, pp. 26-35, Springer.
- 4 Bingham, E. & Mannila, H. 2001. 'Random projection in dimensionality reduction: applications to image and text data'. *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining (KDD 2001)*, August 26-29 2001, San Francisco, CA, USA. ACM, pp. 245 - 250.
- 5 Cil, I. 2012. 'Consumption universes based supermarket layout through association rule mining and multidimensional scaling'. *Expert Systems with Applications* 39(10), pp. 8611-8625.
- 6 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: Meersman, R., Tari, Z., Schmidt, D. C. (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* 2888, pp. 389-406, Springer
- 7 de Medeiros, A. K. A. & Weijters, A. J. M. M. 2005. 'Genetic Process Mining'. *Lecture Notes in Computer Science* 3536, pp. 48–69, Springer.
- 8 Duda, R. O., Hart, P. E. & Stork, D. G. 2000. *Pattern Classification (2nd ed.)*. John Wiley and Sons, New York.
- 9 Goedertier, S., Weerd, 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), pp. 1697-1710.
- 10 Goldberg, K., Roeder, T., Gupta, D. & Perkins, C. 2001. 'Eigentaste: A constant time collaborative filtering algorithm'. *Information Retrieval Journal* 4(2), pp. 133–151.
- 11 Golub, G. H. & Reinsch, C. 1970. Singular value decomposition and least squares solution. *Numerische Mathematik* 14(5), pp. 403-420.
- 12 Gong, Y. & Liu, X. 2000. 'Video Summarization using Singular Value Decomposition'. *2000 Conference on Computer Vision and Pattern Recognition (CVPR 2000)*, June 13-15 2000, Hilton

- Head, SC, USA. *IEEE Computer Society* 1, pp. 174-180.
- 13 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), pp. 1010-1027.
  - 14 Günther, C. W. & van der Aalst, W. M. P. 2007. 'Fuzzy Mining – Adaptive Process Simplification Based on Multi-Perspective Metrics'. In: Alonso, G., Dadam, P., Rosemann, M. (eds), *Business Process Management, 5th International Conference (BPM 2007)*, September 24-28 2007, Brisbane, Australia, Proceedings. *Lecture Notes in Computer Science* 4714, pp. 328-343, Springer.
  - 15 Jagadeesh Chandra Bose, R. P. & van der Aalst, W. M. P. 2009. 'Context Aware Trace Clustering: Towards Improving Process Mining Results'. *Proceedings of the SIAM International Conference on Data Mining (SDM 2009)*, April 30 - May 2 2009. Sparks, Nevada, USA. pp. 401-412.
  - 16 Jain, A. K. & Dubes, R. C. 1988. *Algorithms for Clustering Data*, Prentice-Hall Inc. Englewood Cliffs.
  - 17 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), pp. 13351-13359.
  - 18 Jeong, S., Kim, S. W., Kim, K. & Choi, B. U. 2006. 'An Effective Method for Approximating the Euclidean Distance in High-Dimensional Space'. In: Bressan, S., Küng, J., Wagner R. (eds), *Database and Expert Systems Applications 17th International Conference (DEXA 2006)* September 4-8 2006. Kraków, Poland, Proceedings. *Lecture Notes in Computer Science* 4080, pp. 863-872, Springer.
  - 19 Johnson, W. B. & Lindenstrauss, J. 1984. 'Extensions of lipshitz mapping into Hilbert space'. *Contemporary Mathematics* 26, 189-206.
  - 20 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'. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics (SMC 2011)*, October 9-12 2011. Anchorage, Alaska, USA, pp. 1181-1186.
  - 21 Liu, J. & Kavakli, M. Year. 'Hand Gesture Recognition Based on Segmented Singular Value Decomposition'. In: Setchi, R., Jordanov, I., Howlett, R. J. & Jain, L. C. (eds), *Knowledge-Based and Intelligent Information and Engineering Systems - 14th International Conference (KES 2010)*, September 8-10 2010. Cardiff, UK. pp. 214-223.
  - 22 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), pp. 424-437.

- 23 MacQueen, J. 1967. 'Some Methods for Classification and Analysis of Multivariate Observation'. *Proc. of the 5th Berkeley Symp. on Mathematical Statistics and Probability*, pp. 281-297, University of California Press.
- 24 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: In L. Azevedo and 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.
- 25 Markos, A. I., Vozalis, M. G. & Margaritis, K. G. Year. 'An Optimal Scaling Approach to Collaborative Filtering Using Categorical Principal Component Analysis and Neighborhood Formation'. In: Papadopoulos, H., Andreou, A. S. & Bramer, M., (eds), *Artificial Intelligence Applications and Innovations (AIAI 2010)*, October 6-7 2010. Larnaca, Cyprus, Proceedings. *IFIP Advances in Information and Communication Technology* 339, pp. 22-29, Springer.
- 26 Maruster, L. & Beest, N. R. T. P. v. 2009. 'Redesigning business processes: a methodology based on simulation and process mining techniques'. *Knowledge Information Systems* 21, pp. 267-297.
- 27 Megalooikonomou, V., Li, G. & Wang, Q. 2008. 'A dimensionality reduction technique for efficient time series similarity analysis'. *Information Systems* 33(1), pp. 115-132.
- 28 Meulman, J., van der Kooij, A. & Heiser, W. 2004. *Principal components analysis with nonlinear optimal scaling transformations for ordinal and nominal data*. In: Kaplan, D. (eds), *Handbook of Quantitative Methods in the Social Sciences*. Sage Publications, Newbury Park.
- 29 Nicholas, C. K. & Dahlberg, R. Year. 'Spotting Topics with the Singular Value Decomposition'. In: Munson, E. V., Nicholas, C. K. & Wood, D. (eds), *Principles of Digital Document Processing, 4th International Workshop (PODDP'98)*, March 29-30 1998. Saint Malo, France, Proceedings. *Lecture Notes in Computer Science* 1481, pp. 82-91, Springer.
- 30 Pelleg, D. & Moore, A. W. 2000. 'X-means: Extending K-means with Efficient Estimation of the Number of Clusters'. In: Langley, P. (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.
- 31 Process Mining Group, Math&CS department, Eindhoven University of Technology. 2009. <http://www.processmining.org/prom/start>
- 32 Reijers, H. A., Song, M. & Jeong, B. 2009. 'Analysis of a collaborative workflow process with distributed actors'. *Information Systems Frontiers* 11(3), pp. 307-322.
- 33 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'. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (RSMC)* 39, pp. 474-479.



- 34 Rozinat, A. & van der Aalst, W. M. P. 2008. 'Conformance Checking of Processes Based on Monitoring Real Behavior'. *Information Systems* 33(1), pp. 64-95.
- 35 Sano, A. 1993. 'Optimally regularized inverse of singular value decomposition and application to signal extrapolation'. *Signal Processing* 30, pp. 163-176.
- 36 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.
- 37 Shlens, J., 2005. 'A Tutorial on Principal Component Analysis'. *Institute for Nonlinear Science, UCSD*, April 22, 2009. Version 3.01.
- 38 Song, M. & van der Aalst, W. M. P. 2008. 'Towards Comprehensive Support for Organizational Mining'. *Decision Support Systems* 46(1), pp. 300-317.
- 39 Song, M., Gunther, C. W. & van der Aalst, W. M. P. 2008. 'Trace Clustering in Process Mining'. In: Ardagna, D., Mecella, M., Yang, J. (eds), *Business Process Management Workshops (BPM 2008)*, September 1-4 2008. Milano, Italy, *Lecture Notes in Business Information Processing* 17, pp. 109-120, Springer.
- 40 Tan, P.-N., Steinbach, M. & Kumar, V. 2006. *Introduction to Data Mining*, Boston, MA, USA,, Pearson Addison Wesley.
- 41 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), pp. 54-66.
- 42 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. & Verbeek, H. M. W. 2007. 'Business Process Mining: An Industrial Application'. *Information Systems* 32(5), pp. 713-732.
- 43 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, pp. 3-21.
- 44 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), pp. 1128-1142.
- 45 Wall, M., Rechtsteiner, A. & Rocha, L. M. 2003. 'Singular value decomposition and principal component analysis'. In: Berrar, I. D. P., Dubitzky, W. & Granzow, M. (eds), *A Practical Approach to Microarray Data Analysis*, Springer, Kluwer, Norwell, MA.
- 46 Weijters, A., van der Aalst, W. M. P. & de Medeiros, A. K. A. 2006. 'Process mining with the heuristics miner algorithm'. *BETA Working Paper Series WP 166*, Eindhoven University of Technology, Eindhoven.
- 47 Witten, I. H., Frank, E. & Hall, M. A. 2011. *Data Mining: Practical Machine Learning Tools and*

*Techniques*, Morgan Kaufmann Publishers Inc., San Francisco.

- 48 Xu, X. & Wang, X. Year. 'An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines'. In: Li, X., Wang, S. & Dong, Z. Y. (eds), *Advanced Data Mining and Applications, First International Conference (ADMA 2005)*, July 22-24, 2005. Wuhan, China, Proceedings. *Lecture Notes in Computer Science* 3584, pp. 696-703, Springer.
- 49 Ying, C. L. & Jin, A. T. B. 2007. 'Probabilistic Random Projections and Speaker Verification'. In: Lee, S.-W. & Li, S. Z. (eds), *Advances in Biometrics, International Conference (ICB 2007)*. August 27-29 2007. Seoul, Korea, Proceedings. *Lecture Notes in Computer Science* 4642, pp. 445-454, Springer.
- 50 Zhao Zhang, M. J., Ning Ye 2010. 'Effective multiplicative updates for non-negative discriminative learning in multimodal dimensionality reduction'. *Artificial Intelligence Review* 34(3), pp. 235-260.
- 51 Zho, Y. & Karypis, G. 2005. 'Hierarchical Clustering Algorithms for Document Datasets'. *Data Mining and Knowledge Discovery* 10(2), 141-168.

## **Acknowledgement**

First of all, I would like to thank my advisor, Dr. Minseok Song, for his invaluable advice, his constant support, and tireless encouragement. My frequent interactions with Dr. Minseok Song were also very valuable learning experiences. Another really great thanks goes to my thesis committee members, Dr. Hangyun Woo and Dr. Duck Young Kim for serving on my thesis committee and providing me many structural comments and recommendations. Also, I gratefully acknowledge all professors here at the School of Technology Management, Dr. Minseok Song, Dr. Kwanho Kim, Dr. Hyeongsop Shim, Dr. Sangdo Oh, Dr. Yeong-Ho Woo, Dr. Han-Gyun Woo, Dr. Kyootai Lee, Dr. Dongryul Lee, Dr. Jinhyouk Im, Dr. Young Bong Chnag, Dr. Keunsuk Chung, Dr. Yoonhyuk Jung, Dr. Hyewook Jeong, Dr. WooJe Cho, Dr. Insook Cho, and Dr. Sang-Tai Choi, for their guidance and encouragement.

Furthermore, I would like to thank my colleagues at my department and lab for supporting me academically and emotionally throughout two years. In particular the colleagues from Dr. Song's lab, Jason, Yonghyeok, Sookyoung, Miae, Hossein, and Minjeong who had the patience to listen to me explaining the status of my research and who reacted always in a motivating way. Also I wish to express my gratitude to Juyeon, Euijoo, Hyunjoo, JiYu, Jaeha, Chaerin, and Seoye for supporting me in my lowest times and spending pleasant time together. And I great thanks goes to Young Sim, and Jeong-il for helping me adjust to the new surroundings at my first semester.

Finally, I wish to express my gratitude to my family for all the years of encouragement and mental support. This thesis is dedicated to them.

