http://dx.doi.org/10.5392/IJoC.2012.8.1.023

Unsupervised Clustering of Multivariate Time Series Microarray Experiments based on Incremental Non-Gaussian Analysis

KamSwee Ng
Department of ATTD Automation
Intel Products (M), Hi-Tech Park Kulim 09000, Kedah Malaysia

Hyung-Jeong Yang*, Soo-Hyung Kim
Department of Computer Science
Chonnam National University, Gwangju 500-757, South Korea

Sun-Hee Kim
Department of Computer Science
Carnegie Mellon University, Pittsburgh 15213, USA

Nguyen Thi Ngoc Anh
Department of Electronics and Computer Engineering
Chonnam National University, Gwangju 500-757, South Korea

ABSTRACT

Multiple expression levels of genes obtained using time series microarray experiments have been exploited effectively to enhance understanding of a wide range of biological phenomena. However, the unique nature of microarray data is usually in the form of large matrices of expression genes with high dimensions. Among the huge number of genes presented in microarrays, only a small number of genes are expected to be effective for performing a certain task. Hence, discounting the majority of unaffected genes is the crucial goal of gene selection to improve accuracy for disease diagnosis. In this paper, a non-Gaussian weight matrix obtained from an incremental model is proposed to extract useful features of multivariate time series microarrays. The proposed method can automatically identify a small number of significant features via discovering hidden variables from a huge number of features. An unsupervised hierarchical clustering representative is then taken to evaluate the effectiveness of the proposed methodology. The proposed method achieves promising results based on predictive accuracy of clustering compared to existing methods of analysis. Furthermore, the proposed method offers a robust approach with low memory and computation costs.

Keywords*: Multivariate Time Series, Principle Component Analysis, Independent Component Analysis Article, Microarray Analysis, Feature Selection, Incremental Model, Clustering.*

1. INTRODUCTION

Time series microarray experiments have become an indispensable technology, which allows for the monitoring of expression levels of thousands genes under a variety of conditions.

A microarray is typically a glass slide on to which DNA molecules are fixed in an orderly manner at specific locations called features. A microarray may contain thousands of features and each feature may contain a few million copies of identical DNA molecules that uniquely correspond to a gene. Therefore, the high dimensionality of microarray gene expression is always a challenge for biologists to obtain comprehensive insights on microarray data [1]. The unique nature of microarray data, which contains a high number of features but a relatively small number of observations, is in contrast to the normal behavior of most real world data. Furthermore, the gene

* *Corresponding author: E-mail: hjyang@jnu.ac.kr*
Manuscript received Jan. 09, 2012; accepted Feb.17, 2012

** This research was supported by the MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Informati on Technology Research Center) support program supervis ed by the NIPA(National IT Industry Promotion Agency) (NIPA-2012-H0301-12-3005). This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Minist ry of education, Science and Technology (2011-0003693).*

expressions of two individuals are rarely the same. Genetic variation poses more challenges to analysis of microarray data. In addition, while the microarray experiment is carried out, the data tends to be noisy as a result of tissue collection and amplification of messenger RNA (mRNA) to hybridization onto the chip [2]. Time series microarray experiments are being popularly used to characterize the dynamic biological processes of genes across time. Its ability to analyze a huge number of genes in one experiment has encouraged biologists to collect more samples in microarray data [3]. Therefore, it is expected that more patients' data may become available in the future. With advances in modern technology, more genes may also be discovered by biologists. Hence, both the number of samples and genes are expected to grow and thus, the size of microarray data will increase. In this case, an appropriate method should be proposed to handle the problem of huge data dimensions by retaining the important information and eliminating noise.

Since, among the huge number of genes presented in a microarray, only a small number of genes are expected to be effective for performing a certain task. For delivering precise, reliable and interpretable results, it is desirable to identify a small subset of genes in developing gene expressions. Therefore, gene selection is the main goal to discover a reduced set of the most relevant genes. For multivariate microarray analysis, several techniques have been developed, because it is important to seek results that take into account the relationships between multiple variables, as well as within the variables [5]. Multivariate analysis is widely used to extract features and reduce the dimension of microarray datasets.

The most common multivariate analyses in dimensionality reduction are known as Principal Component Analysis (PCA) and Independent Component Analysis (ICA). These are used to summarize the time series microarray data. ICA is a method in multivariate statistical analysis used to separate data into underlying informational components [3], [6]. ICA is essentially a useful method to reveal the driving forces that underlie a set of observed phenomena. These phenomena include the firing of a set of neurons from microarray datasets. ICA has been applied to many goals, such as separation of artifacts in magnetoencephalogram (MEG) data, as well as visualization, localization, and feature extraction of the electroencephalogram (EEG) signal [7–10].

PCA projects the data to a new space in a lower dimension to capture the data with the highest variance spanned by the orthogonal principle components. This means that PCA finds a set of signals with a much weaker property than independence, while ICA finds a set of independent source signals [6], [22]. Specifically, PCA finds a set of signals that are uncorrelated to each other. In some cases, if the data are Gaussian, estimation of the model requires an orthogonal transformation [11]. However, PCA suffers from its orthogonality requirement for real world data whose distribution is not Gaussian. In probability theory, the Central Limit Theorem (CLT) states conditions under which the sum of a sufficiently large number of independent random variables, each with finite mean and variance, will be approximately normally distributed [12]. That is, mixtures of several sources tend to be more Gaussian than the distribution of the original sources [13]. PCA does provide a set of independent components but only if those components are Gaussian. Conversely, ICA is considered as a non-Gaussian factor analysis, in which ICA decomposes the statistical independent components. Many studies have shown that ICA outperforms PCA in microarray analysis.

Many applications dealing with massive microarray datasets are emerging. It is necessary to analyze the data as soon as the data arrives [7], [14]–[16]. Unfortunately, the traditional way of processing microarray datasets always treats these data as static. Besides, batch processing, especially in time series data requires time that depends on the duration t, which grows to infinity. Both classical PCA and ICA involve the calculation of Singular Value decomposition (SVD), which consumes huge amounts of memory [17]. As the space requirement also depends on t, the consumption of space is proportional to the duration t. Thus, batch mode processing always suffers from the large memory requirement and is time consuming, especially when the size of the data increases. On the fly processing is desirable to efficiently process the data. The incremental learning model is therefore proposed as a better alternative to process the data with less memory and time consumption. The incremental model works by processing the data at each input vector while the historical data are stored in a few variables. The previous values of the variables are updated by the next input vector [17]–[21]. This process is repeated until the end of the input data. Since historical values are kept in the variables, the incremental model has only a small memory requirement and thus accelerates the processing speed.

In this paper, we propose a method that integrates ideas based on an incremental approach and ICA. The proposed method computes an orthogonal weight by updating each weight vector in a predefined energy range. Upon obtaining the orthogonal weight, it is converged to become a non-Gaussian weight. This methodology works incrementally by updating the non-Gaussian weight using past variables instead of re-computing the entire dataset when new data input arrives; thus, it has a low computation cost. Our proposed method adapts the concepts of converging statistically independent components of multivariate data in an incremental way. The traditional batch method is limited by the requirement of re-computation of the entire data matrix when there is new input data. In contrast, our proposed method can efficiently integrate newly arriving data with past variables without involving the entire data matrix.

The remainder of this paper is organized as follows. Section 2 reviews the principal of ICA and our proposed methodology. The experimental results are presented in Section 3. Finally, conclusions and future work are outlined in Section 4.

2. PROPOSED METHOD

In this chapter, the fundamental concept of ICA is first presented. Then, a newly proposed method is modeled in order to enhance the way in which the idea of ICA is integrated incrementally to successful analysis of multivariate microarray time series data.

2.1 Fundamentals of ICA Model for Microarrays

The data from microarray experiments can be formed by large matrices of expression levels of genes (n columns) under different conditions (m rows). Time series microarray experiments can be represented in the ICA model by the following equation:

$$x = As \tag{1}$$

Where bold lower-case letters indicate vectors and bold upper-case letters denote matrices. $x = [x_1, x_2, ..., x_n]$ is an *mxn* microarray gene expression matrix of *t* experiments and *n* genes. *A* is the mixing parameter and *s* is the hidden variable. To precisely illustrate the ICA mixing model, Equation 1 can be expanded as follows:

$$\begin{bmatrix} x_1(t) \\ x_2(t) \\ \vdots \\ x_n(t) \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \ddots & a_{21} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n1} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} s_1(t) \\ s_2(t) \\ \vdots \\ s_n(t) \end{bmatrix} \qquad (2)$$

Where $a_{11}, a_{12,} a_{13}, ..., a_{nn}$ are some parameters representing the controlling factors. $s_1(t), s_2(t), s_3(t)...s_n(t)$ are the biological processes that form the evolving base of the experiments. This can be summarized in that x is the sum of the amount of controlling factors, A weighted by the biological process, s. We denote the weight, W as the inverse matrix of mixing parameter, A and thus the hidden variables, s can be found using the expression. ICA finds *A* such that the source signals, *s* or components are statistically independent. After estimating the matrix *A* , we can compute its inverse, *W* to obtain the independent components as shown below:

$$s = A^{-1}x \qquad (3)$$

$$s = Wx \qquad (4)$$

The model can be also written as:

$$s = \sum_{i=1}^{n} w_i x_i \qquad (5)$$

Assumptions on the ICA model are that the components *s* is statistically independent and that the independent components must have non-Gaussian distribution. Consider *y*, a linear combination of s_i . We seek one of the independent components, *y*:

$$y = w^T x \qquad (6)$$

By maximizing the non-Gaussianity of $w^T x$, we can obtain the independent component y.

2.2 Incremental Non-Gaussian Analysis for Microarrays

Incremental non-Gaussian analysis for microarray expression, which integrates an incremental model with the concept of ICA to update the non-Gaussian weight, is presented here. Table 1 shows the description of the symbols used in the proposed method.

Table 1. Notation

Symbol	Description
x	Input vector (lower-case bold)
x_t	The n stream input values at time t
W	Weight matrix (upper-case bold)
w_i	The i-th participation weight vector
n	Number of streams
k	Number of hidden variables
y_t	Vector of hidden variables for x_t
d_i	Energy estimate of i-th hidden variable
e_i	Reconstruction error
$E_{t,i}$	Total energy of hidden variables up to time t , and hidden variable i
E_{hv}	Total energy of hidden variables
E_x	Energy of input data x
λ	Exponential forgetting factor
f	Lower bound predefined energy
F	Upper bound predefined energy

The propose method will be described as follows. In the time series microarray data, $x_t \in \Re^n$ is the n genes measurement column-vector of each experimental sample time, t that might grow continuously to infinity. In the first experiment, the basis vector is adopted by weight vector, w_i . Each of the weight vectors w_i is projected onto the input vector, x_t in the linear transformation of the data stream to obtain the hidden variables or components, y_t over time. The core idea of the incremental approach is to gradually update each of the participation weight vectors, w_i at each time tick in the newly projected space. The weight vector of the proposed method is non-Gaussian. Upon obtaining the orthogonal weight vector, w_i from the incremental model, each of the weight vectors, w_i are updated until the maxima of non-Gaussianity is obtained.

Firstly, the number of hidden variables, *k* is initialized with an arbitrary number. Then, we obtain the input vector, $x_t = \left[x_{t,1}, ... x_{t,n}\right]^T$ at time, *t* with *n* dimensions. From the input vector, we compute the *i-th* component, $y_{t,i}$ based on the previous weight, $w_{t-1,i}, 1 \le i \le k$. The computation of the *i-th* component is shown in the following equation. It is computed by the sum of the weight vector projected onto the input vector at time, *t*:

$$y_{t,n} = \sum_{n=1}^{n} W_{i,n} X_{t,n} \qquad (7)$$

Next, we estimate the energy, $d_i = \lambda d_i + y_i^2$ and the reconstruction error, $e_i = \tilde{x}_i - x_i$ based on the hidden variable calculated from the previous step. The initial value of energy is set to a small positive value.

The exponential forgetting factor, λ is introduced so that new data can adapt to previous behavior in the data stream. The value of the exponential forgetting factor is between 0 and 1. The introduction of the λ value helps to reduce the huge memory usage, because there is no buffer space requirement for the whole data stream. The common choices of the exponential forgetting factor are values between 0.96 and 0.98. The exponential forgetting factor should be set to a high value, so that the data can adapt to the past values [18]. As long as the value does not vary too much, the result is similar. The magnitude of the estimates should also consider the past data captured by the participation weight vector, w_i . For this reason, the update is inversely proportional to the current energy, $E_{t,i}$ of the *i-th* hidden variable; that is $E_{t,i} = \frac{1}{t}\sum_{\tau=1}^{t} y_{\tau,i}^2$ and $d_i = tE_{t,i}$. The participation weight vector is updated based on the following equation:

$$w_i = w_i + \frac{y_i e_i}{d_i} \qquad (8)$$

Finally, we obtain the updated participation weight, $w_i, 1 \le i \le k$. The actual hidden variables, y_t at time *t,* are computed by projecting the weight matrix, *w* with the input vector, *x*.

To make sure that there are sufficient components to represent the data, energy thresholding is applied to determine how many hidden variables are needed. The energy retained by the hidden variables, E_{hv} is compared with the upper, $F_E E$ and lower bound energy, $f_E E$ of the original input data. If the hidden variables maintain too little energy, the number of hidden variables are increased, *k*. Conversely, if the maintained energy is too high, the number of hidden variables, *k,* will be decreased. This ensures that the energy of the hidden variables is always within the predefined specified interval of low and high energy values. Whenever a new datum arrives, the process of updating the weight vector will be repeated and the number of hidden variables will be adjusted to retain the energy of the components between the predefined low and high energy bounds. The algorithm is shown in Figure 1 below.

```
Input:
New Input $\boldsymbol{x} \in \mathbb{R}^n$
Old weight matrix $\boldsymbol{W} \in \mathbb{R}^{k\times n}$
Old energy estimate by past data $\boldsymbol{d} \in \mathbb{R}^k$
Old total hidden variable energy $E_{hv}$
Old total input data energy $E_x$
Output:
Updated weight matrix $\boldsymbol{W} \in \mathbb{R}^{k\times n}$
Updated energy estimate $\boldsymbol{d} \in \mathbb{R}^k$
Updated total hidden variable energy $E_{hv}$
Updated total input data energy $E_x$

for $i = 1$ to k // k is number of hidden variables
  $y_i = \mathbf{w}_i^T \mathbf{x}$ //project $\mathbf{x}$ onto weight vector $\mathbf{w}_i$
  $d_i = \lambda d_i + y_i^2$ //estimate energy captured by past data
  $e_i = \mathbf{x}_i - y_i \mathbf{w}_i$ //estimate the residue error
  $\mathbf{w}_i = \mathbf{w}_i + \frac{1}{d_i} y_i e_i$ // update component estimate)
  while not convergence
    $\mathbf{w}_i = E\{\mathbf{x}_i g(y_i)\} - E\{g'(y_i)\} \times \mathbf{w}_i$ //maximize non-Gaussianity
  end
end
$\mathbf{y} = \mathbf{W}^T \mathbf{x}$ //compute hidden variables
$E_{hv} = \lambda E_{hv} + \mathbf{y}^2$ //compute total hidden variables energy
$E_x = \lambda E_x + \mathbf{x}^2$ //compute total input data energy
if $E_{hv} < f E_x$
  $k = k + 1$ //add one more hidden variable
else if $E_{hv} > F E_x$
  $k = k - 1$ //deduct one hidden variable
end
```

Fig. 1. Non-Gaussian analysis algorithm snippet

3. EXPERIMENTAL RESULTS

3.1 Datasets

To evaluate the effectiveness of our proposed method in the microarray dataset, the Gene Expression Omnibus (GEO datasets deposited by Blalock are utilized for the experiment [19]. The datasets are studied to analyze the hippocampal gene expression in the control and Alzheimer's Disease (AD) of varying severity on 31 dedicated microarrays. The samples are obtained from the Brain Bank of the Alzheimer's Disease Research Center at the University of Kentucky. Human GeneChips (HG-U133A) and Microarray Suite 5 are used for data collection. The samples with significant noise are eliminated, leaving eight control and five severe AD samples. The unregulated genes in microarray data often contain little information, thus these unregulated genes are removed from the experiment. Therefore, we have 13 samples and 3617 genes for each sample.

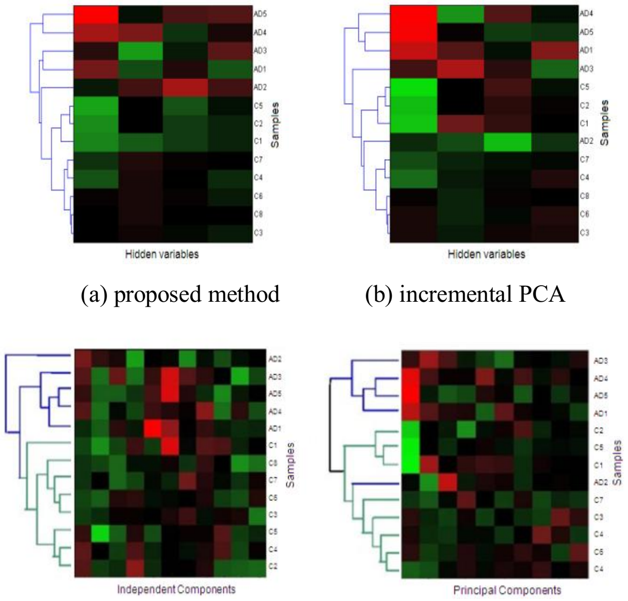
3.2 Clustering of Microarray Datasets

Our proposed method decomposes the non-Gaussian weight vectors that are statistically independent. The underlying biological processes of the microarray gene expression data are more super-Gaussian than the mixture of the original sources. Only a small number of genes are expected to be changed at each pathological transition [8]. This leaves the majority of genes unaffected. Therefore, it forms a super-Gaussian distribution. Non-Gaussian analysis is thus suitable to analyze the microarray gene expression data. We perform unsupervised hierarchical clustering on the AD microarray dataset. Let the microarray data, *X (m,n)* be a two-dimensional m by n matrix. Each row, m contains the observation of gene profiles and each column, n shows the genes across the experiments. The data are first normalized to zero-mean and unit variance to standardize the data. This is done by subtracting the data from the mean value and then dividing the value by the standard deviation. Prior to the clustering process, the input vector and gene profiles of each experiment are decomposed into non-Gaussian weight vectors. Then, the non-Gaussian weight vectors are updated at the next input vector (gene profiles at the next experiment). This process is repeated by updating the weight of the gene profiles at each experimental observation until the last observation. Upon completion of the pre-processing on the microarray data, the final non-Gaussian weight vectors are obtained. In this experiment, the forgetting factor is determined to be 0.96 and the energy range from 95 per cent to 98 per cent. From the pre-processing process, the final dimension of the microarray data matrix is reduced to four. The data matrix used for clustering is a 13 by four matrix. Figure 2a shows the clustering result of the projection of the microarray data on the non-Gaussian weight vectors. The control and severe AD samples can be clearly discriminated using a small number of features.

Incremental PCA [17] is adopted to compare other methods in this section. The main difference between our proposed method and incremental PCA is that the decomposed components are not independent. That is, incremental PCA is similar to PCA, except that it works in an incremental fashion. The experiment is conducted in the same way for our proposed method and with the same parameter setting. Figure 2b shows the clustering result. It is evident that one AD sample, AD2 cannot be clustered correctly.

The experimental results are compared to the ICA result by Kong [16]. Figure 2c illustrates the clustering result by ICA. The entire data matrix is conducted by fastICA algorithm [8]. FastICA is repeated 50 times to alleviate the instability of the slightly different results generated from each looping. Eleven ICA latent variables are identified to sufficiently capture the significant underlying biologically information from the original data matrix. Both the control and AD severe samples can be discriminated into correct clusters. However, comparing this with our proposed method, the proposed method achieves promising results with fewer components.

PCA is a linear projection in the sense that the variables of the projection space are linear combinations of the gene expressions. In the PCA case, the result is also obtained from Kong [16]. The experiment is carried out by decomposing the gene expression matrix into principal components by preserving 95.5 per cent of the variance. The principal components with low variance that contained noise are removed from the clustering process. Figure 2d demonstrates the clustering result of PCA. The control samples are successfully clustered, but in the case of AD samples the AD2 sample is clustered incorrectly.

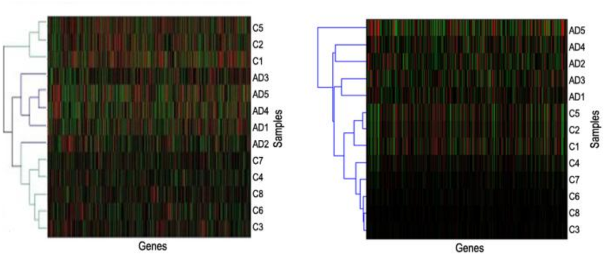


(a) proposed method (b) incremental PCA

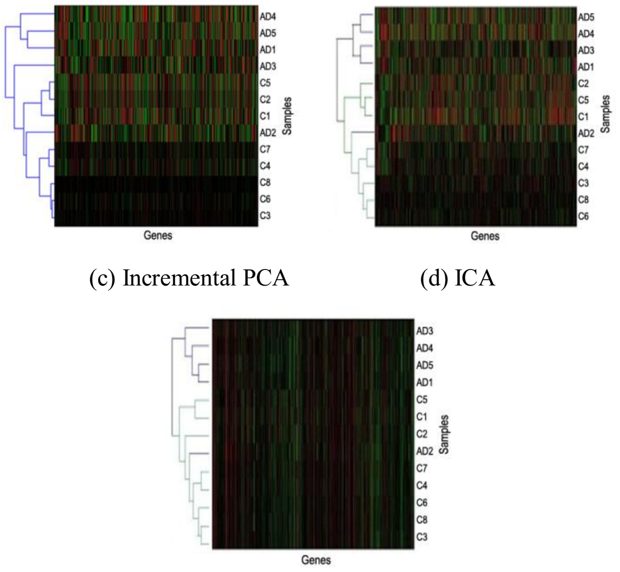
(c) ICA (d) PCA

Fig. 2. Hierarchical clustering of proposed method (a), incremental PCA (b), ICA (c), and PCA (d) outputs.

We examine the abilities of the methods above in discriminating the AD samples from the control samples after the reconstruction. The reconstructed data are obtained by projecting the raw data into the latent variables found in these methods. A comparison is made against the hierarchical clustering results performed on the normalized raw data and the reconstructed data by the methods above. Figure 3a displays the clustering result of the normalized raw data. Some of the AD samples are clustered together, but the hierarchy of the cluster does not discriminate the two different clusters successfully.



(a) Normalized raw data (b) Proposed method



(c) Incremental PCA (d) ICA

(e) PCA

Fig. 3. Hierarchical clustering of the normalized raw data (a), data reconstructed by proposed method (b), incremental PCA (c), ICA (d), and PCA (e).

For our proposed method, the data are reconstructed by projecting the raw data into the four dimensional non-Gaussian weight vectors obtained by incrementally updating the weight vectors at each observation using 0.96 as the exponential forgetting factor and an energy range of 95 percent to 98 percent. Figure 3b shows the clustering result applied to the reconstructed data by our proposed method. It is evident that the control and AD samples are separated into different groups. The proposed method can improve the discriminative ability of the clustering result.

Figure 3c depicts the clustering result of the reconstructed data by incremental PCA. The data is reconstructed by projecting the raw data into the orthogonal weight vectors obtained from incremental PCA. The parameter setting is the same as for our proposed method, to achieve fair comparison. The result demonstrates that incremental PCA fails to separate one AD sample, AD2, from the control samples.

In the PCA and ICA method, ten principal components captured 95.5 per cent of the variance and eleven independent components, which are identified as being involved in biological processes, are selected to reconstruct the data respectively. One AD sample, AD2, is not clustered into the AD group correctly by PCA and ICA [16]. Although more components are used for reconstruction, the proposed method shows a more promising clustering result with fewer components than those of other methods.

3.3 Qualitative Evaluation

This section compares the qualitative performance of our proposed method with that of ICA. ICA is chosen for the comparison with respect to our proposed method because both algorithms decompose non-Gaussian components. In this section, experiments are conducted on the AD microarray dataset because they have different behaviors in multivariate data. Synthetic data are augmented on these datasets so that more features and more observations are generated. Figure 4 shows the plot of execution time against number of genes on the microarray dataset. The proposed method is plotted with star symbols, whereas ICA is plotted with plus signs. In the microarray experiments, the number of observations and the other parameters setting are fixed. However, the number of genes increases in each loop, so that the execution time of different number of genes in both our proposed method and ICA can be recorded. The exponential forgetting factor is set at 0.96, the energy range is from 95 to 98 per cent and there are initially three hidden variables. When the number of genes increases, the execution times of both methods proportionally increase. However, from the graph, it is evident that ICA requires more computation time than our proposed method when the number of genes increases.

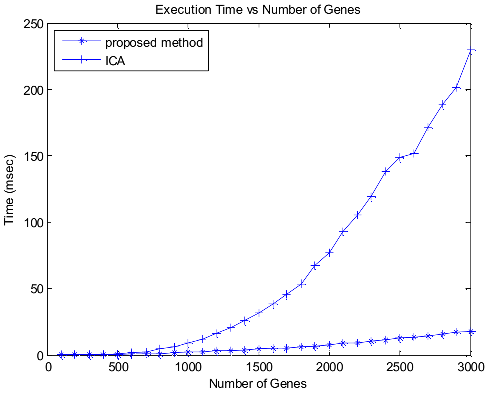


Fig. 4. Plot of execution time against number of genes on the microarray dataset

As shown in Fig. 4, the proposed method only involves floating operation, whereas ICA involves covariance matrix calculation. ICA requires a longer computation time when the size of the matrix increases. Thus, our proposed method is proven more efficient when the number of genes or sources is growing.

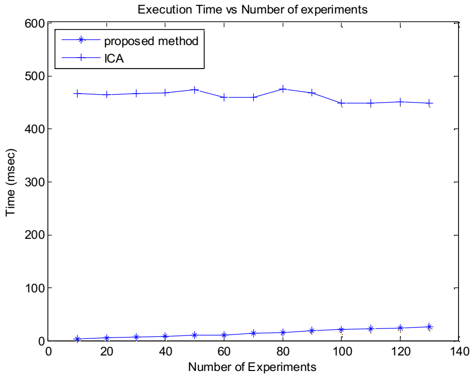


Fig. 5. Plot of execution time against number of experiments in microarray data.

Fig. 5 illustrates the plot of execution time against the number of experiments in the microarray dataset. The parameters are the same as in the previous experiment, except that the number of genes is fixed in this experiment. The number of experiments is varied in each iteration, so that the execution time of different experiment sizes can be observed. When the number of experiments increases, the execution time does not increase proportionally since, when a new experiment contains gene expression; our algorithm updates weight vectors with a new input vector using the past value stored in variables. However, when the number of experiments increases, the plot for ICA shows a dramatic upwards trend. Further, the execution time of ICA is also higher than that of our proposed method.

The graph in Fig. 5 shows that ICA takes a long time to execute. This can be explained by ICA's involving a covariance matrix calculation, requiring high levels of memory and computation time. In addition, when a new experiment is available, ICA requires re-computation of the entire microarray matrix. Therefore, more computation is required. It is proven that our proposed approach is robust in both the number of features (gene) and number of observations. Hence, it is suitable for microarray analysis.

4. CONCLUSION AND FUTURE WORK

It has been shown that an incremental non-Gaussian model can be used to effectively cluster data for microarray expression matrices. In the microarray dataset, the super-Gaussian weight matrix reveals the underlying biological processes in the microarray data. The clustering accuracy for the microarray shows promising results using our proposed method when compared with previous analysis such as ICA or PCA.

Furthermore, the proposed method demonstrates a robust approach, with low memory and computation costs. It scales linearly with the stream size, number of sources, and hidden variables. In contrast, both ICA and PCA have limitations on computation power when the number of genes and observations grows larger. They are also constrained to re-compute the entire microarray matrix when a new experiment is added.

For future work, the proposed method can be investigated in multi-way data analysis. Analyzing multi-way data can reveal more behavior by discovering the correlation between different dimensions. We can capture a multi-linear structure using higher order statistics incrementally. If the data consists of more than two modes, the underlying structures can be detected more efficiently using our incremental approach.

REFERENCES

[1] M. Madan Babu, "Introduction to microarray data analysis," in Computational Genomics Horizon Press, U.K, 2009, pp. 225-249.

[2] B. Xie, W. Pan and X. Shen, "Penalized mixtures of factor analyzers with application to clustering high dimensional microarray data," in Bioinformatics, 2009, pp. 501-508..

[3] S. Raychaudhuri, J. M. Stuart and R. B. Altman, "Principal component analysis to summarize microarray experiments: application to sporulation Time Series'" in Pacific Symposium on Biocomputing, 2000, pp. 452-463.

[4] A. L. Boulesteix, C. Porzelius and Martin Daumer, "Microarray-based classification and clinical predictors: on combined classifiers and additional predictive value," in Bioinformatics, 2008, pp. 1698-1706.

[5] C. Das, P. Maji and S. Chattopadhyay, "Supervised gene clustering for extraction of discriminative features from microarray data," in India Conference (INDICON), Annual IEEE, 2010, pp. 1-4.

[6] S. I. Ao and M. K. Ng, “Gene expression time series modeling with principal component analysis,” in Soft Computing, A Fusion of Foundations, Methodologies and Appications, Springer Berlin, February 2006, vol. 10, pp. 351-358.

[7] M. Ungureanu, C. Bigan, R. Strungaru and V. Lazarescu, “Independent component analysis applied in biomedical signal processing,” in Measurement Science Rev, 2004, vol. 4, pp. 1-8.

[8] A. Hyvarinen and E. Oja, “Independent component analysis: algorithms and applications,”, in Neural Network, vol. 13, 2000, pp. 411-430.

[9] M. Dyrholm, “Model selection for convolutive ICA with an application to spatio-temporal Analysis of EEG,” in Neural Computation, vol. 19, 2007, pp. 934-955.

[10] E. Acar, C. A. Bingol, H. Bingol, R. Bro and B. Yener, “Multiway analysis of epilepsy tensors,” in Bioinformatics, vol. 23, 2007, pp. 10-18.

[11] JV. Stone, “Independent component analysis: a tutorial introduction,” in MIT Press, 2004.

[12] Pan JY, H. Kitagawa, C. Faloutsos and M. Hamamoto, “AutoSplit: fast and scalable discovery of hidden variables in stream and multimedia databases,” in Proceedings of the Eighth Pacific Asia Conference on Knowledge Discovery and Data Mining (PAKDD), 2004.

[13] J. L. Semmlow and S. L. Semmlow “Biosignal and biomedical image processing: matlab based application, Marcel Dekker, Inc, 2004.

[14] M. Journee, A. E. Teschendorft, P. A. Absil, S. Tavare and R. Sepulchre, “Geometric optimization methods for the analysis of gene expression data,” in Principal Manifolds for Data Visualization and Dimension Reduction, vol. 58, 2007, pp. 272-292.

[15] L. Zhu and C.Tang “Microarray sample clustering using independent component analysis,” in IEEE /SMC International Conference on System of Systems Engineering, 2006.

[16] W. Kong, X. Mou, Q. Liu, Z. Chen, X. R. Vanderburg, J. T. Rogers and XUdong Huang, “Independent component analysis of Alzheimer’s DNA microarray gene expression data,” in Molecular Neurogenereration, 2009.

[17] S. Papadimitriou, J. Sun and C. Faloutsos, “Streaming pattern discovery in multiple time-series,” in Proceedings of the 31st VLDB Conference, 2009.

[18] J. Sun, , S. Papadimitriou and C. Faloutsos, “Online latent variable detection in sensor networks,” in Proceedings of the 21st International Conference on Data Engineering, ICDE, 2005.

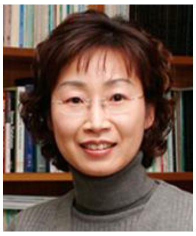
[19] E. M. Blalock, J. W. Geddes, K. C. Chen, N. M. Porter, W. R. Markesbery and P. W. Landfied, “Incipient Alzheimer's disease: Microarray correlation analyses reveal major transcriptional and tumor suppressor responses,” Proceedings of the National Academy of Sciences of the United States of America, vol. 7, 2004, pp. 2173-2178.

[20] R. Mario, S. Cuciniello and D. Feminiano, “Incremental generalized eigenvalue classification on data streams,” in International Workshop on Data Stream Management and Mining, 2005.

[21] K.S Ng, H. J. Yang and S. H Kim,” in BioSystems, vol. 97, 2009, pp. 15-27.

[22] R. Vigario, S. Jaako and O. Erkki, “Searching for independence in electromagnetic brain waves,” in Advances in Independent component analysis, Springer, 2005.

Kam Swee Ng
She received her B.S in University of Technology Malaysia and M.S. in computer science from Chonnam National University. Her current work is Intel Corporation in Malaysia. Her main research interests include data mining, sensor mining and bioinformatics,



Hyung Jeong Yang
She received her B.S., M.S. and Ph. D from Chonbuk National University, Korea. She is currently an associate professor at Dept. of Electronics and Computer Engineering, Chonnam National University, Gwangju, Korea. Her main research interests include multimedia data mining, pattern recognition, artificial intelligence, e-Learning, and e-Design.

Soo Hyung Kim
He received his B.S. at Dept. of Computer Engineering, Seoul National University, and M.S. and Ph.D. at Dept. of Computer Science, Korea Advanced Institute of Science and Technology, Korea. He is currently a professor at Dept. of Electronics and Computer Engineering and a vice-Dean of the Engineering College, Chonnam National University, Gwangju, Korea.

Sun Hee Kim
She received the B.S in Multimedia from Korean Educational Development Institute in 2004 and the M.S. degree in Computer Science from Dongguk University, Korea in 2006. She received the Ph. D. degrees in Computer Science from Chonnam National in 2011. She recently works in Carnegie Mellon University as a researcher. Her research interests include Data Mining, Sensor Mining and Bioinformatic.

Ngoc Anh Nguyen Thi
She received the B.S, in Faculty Mathematics-Informatics from Da Nang Education University, Viet Nam in 2006, and M.S. at Dept. Electronics and Computer Engineering, Chonnam National University, Korea. She is currently a Ph.D. student at Dept. of Electronics and Computer Engineering, Chonnam National University, Korea. From 2006 to 2008, she worked as a lecturer and researcher at Faculty Mathematics-Informatics of Da Nang Education University, Viet Nam. Her research interests focus on the intelligent computing in many applications such as pattern recognitions, bioinformatics, data analysis of data mining and machine learning.