Alternative Model for Extracting Multidimensional Data Based-on Comparative Dimension Reduction

Rahmat Widia Sembiring[1], Jasni Mohamad Zain[2], Abdullah Embong[3]
[1,2]Faculty of Computer System and Software Engineering,
Universiti Malaysia Pahang
Lebuhraya Tun Razak, 26300, Kuantan, Pahang Darul Makmur, Malaysia
[3]School of Computer Science, Universiti Sains Malaysia
11800 Minden, Pulau Pinang, Malaysia
[1]rahmatws@yahoo.com, [2]jasni@ump.edu.my, [3]ae@csusm.my

Abstract. In line with the technological developments, the current data tends to be multidimensional and high dimensional, which is more complex than conventional data and need dimension reduction. Dimension reduction is important in cluster analysis and creates a new representation for the data that is smaller in volume and has the same analytical results as the original representation. To obtain an efficient processing time while clustering and mitigate curse of dimensionality, a clustering process needs data reduction. This paper proposes an alternative model for extracting multidimensional data clustering based on comparative dimension reduction. We implemented five dimension reduction techniques such as ISOMAP (Isometric Feature Mapping), KernelPCA, LLE (Local Linear Embedded), Maximum Variance Unfolded (MVU), and Principal Component Analysis (PCA). The results show that dimension reductions significantly shorten processing time and increased performance of cluster. DBSCAN within Kernel PCA and Super Vector within Kernel PCA have highest cluster performance compared with cluster without dimension reduction.

Keywords: curse of dimensionality, dimension reduction, ISOMAP, KernelPCA, LLE, MVU, PCA, DBSCAN.

1 Introduction

In line with the technological developments, the current data tends to be multidimensional and high dimension, which is complex than conventional data. Many clustering algorithms have been proposed, but for multidimensional data and high dimensional data, conventional algorithms often produce clusters that are less meaningful. Furthermore, the use of multidimensional data will result in more noise, complex data, and the possibility of unconnected data entities. This problem can be solved by using clustering algorithm. Several clustering algorithms grouped into cell-based clustering, density based clustering, and clustering oriented. To obtain an efficient processing time to mitigate a curse of dimensionality while clustering, a clustering process needs data reduction.

Data reduction techniques create a new representation for the data that is smaller in volume and has the same analytical results as the original representation. There are various strategies for data reduction: aggregation, dimension reduction, data compression, discretization, and concept hierarchy [1]. Dimension reduction is a technique that is widely used for various applications to solve curse dimensionality.

Dimension reduction is important in cluster analysis, which not only makes the high dimensional data addressable and reduces the computational cost, but also can provide users with a clearer picture and visual examination of the data of interest [2]. Many emerging dimension reduction techniques proposed, such as Local Dimensionality Reduction (LDR). LDR tries to find local correlations in the data, and performs dimensionality reduction on the locally correlated clusters of data individually [3], where dimension reduction as a dynamic process adaptively adjusted and integrated with the clustering process [4].

Sufficient Dimensionality Reduction (SDR) is an iterative algorithm [5], which converges to a local minimum of $p^* = \arg \min_{\tilde{p} \in P_\theta} D_{KL}[p|\tilde{p}]$ and hence solves the Max-Min problem as well. A number of optimizations can solve this minimization problem, and reduction algorithm based on Bayesian inductive cognitive model used to decide which dimensions are advantageous [6]. Developing an effective and efficient clustering method to process multidimensional and high dimensional dataset is a challenging problem.

The main contribution of this paper is the development of an alternative model to extract data based on density connection and comparative dimension reduction technique. Results of extracting data implemented in DBSCAN cluster, and compare with other clustering method, such as Kernel K-Mean, Super Vector and Random Cluster. This paper is organized into a few sections. Section 2 will present the related work. Section 3 explains the materials and method. Section 4 elucidates the results followed by discussion in Section 5. Section 6 deals with the concluding remarks.

2 Related Work

Functions of data mining are association, correlation, prediction, clustering, classification, analysis, trends, outliers and deviation analysis, and similarity and dissimilarity analysis. Clustering technique is applied when there is no class to predict but rather when the instances divide into natural groups [7, 8]. Clustering for multidimensional data has many challenges. These are noise, complexity of data, data redundancy, and curse of dimensionality. To mitigate these problems dimension reduction needed. In statistics, dimension reduction is the process of reducing the number of random variables. The process classified into feature selection and feature extraction [9], and the taxonomy of dimension reduction problems [10] shown in Fig.1. Dimension reduction is the ability to identify a small number of important inputs (for predicting the target) from a much larger number of available inputs, and is effective in cases when there are more inputs than cases or observations.

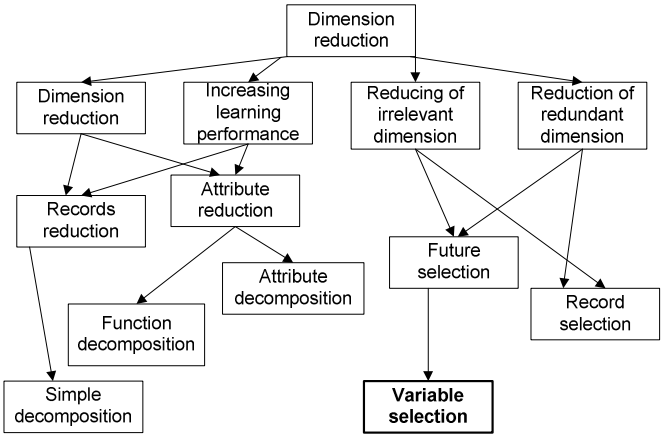

Fig. 1. Taxonomy of dimension reduction problem

Dimensionality reduction techniques have been a successful avenue for automatically extracting the latent concepts by removing the noise and reducing the complexity in processing the high dimensional data [11]. Maaten *et.al* proposed taxonomy dimension reduction technique as shown at Fig. 2, and found traditional dimensionality technique applied PCA and factor analysis, but this technique is unable to handle nonlinear data [12].

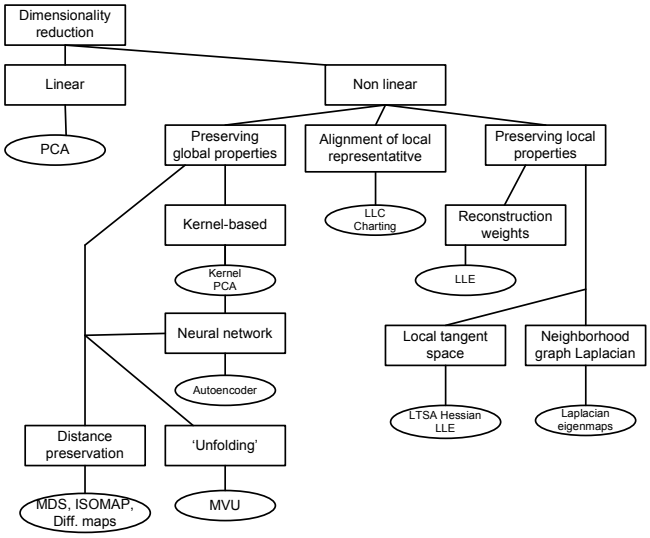

Fig. 2. Taxonomy of dimension reduction technique

The goals of dimension reduction methods are to reduce the number of predictor components and to help ensure that these components are independent. The method designed to provide a framework for interpretability of the results, and to find a

mapping F that maps the input data from the space $\Re^d$ to lower dimension feature space $\Re^d$ denotes as $F(x): \Re^d \rightarrow \Re^{d'}$ [13, 14]. Dimension reduction techniques, such as principal component analysis (PCA) and partial least squares (PLS) can used to reduce the dimension of the microarray data before certain classifier is used [15].

We compared five dimension reduction techniques and embedded in 4 cluster techniques, these dimension reduction are:

A. ISOMAP

ISOMAP (Isometric Feature Mapping) is one of several widely used low-dimensional embedding methods, where geodesic distances on a weighted graph incorporated with the classical scaling. This approach combines the major algorithmic features of PCA and MDS [16, 17] computational efficiency, global optimality, and asymptotic convergence guarantees with the flexibility to learn a broad class of nonlinear manifolds. ISOMAP used for computing a quasi-isometric, low-dimensional embedding of a set of high-dimensional data points. ISOMAP is highly efficient and generally applicable to a broad range of data sources and dimensionalities [18]. ISOMAP Algorithm [16] provides a simple method for estimating the intrinsic geometry of a data manifold based on a rough estimate of each data point's neighbours on the manifold, such as the following phase:

a. Construct neighbourhood graph
   Define the graph G over all data points by connecting points *j and I* if as measured by d_x (i, j), they are closer than e (*e-Isomap*), or if i is one of the K nearest neighbours of j (*K-Isomap*). Set edge lengths equal to $d_x(i, j)$. Determines which points are neighbours on the manifold M, based on the distances dX (i, j) between pairs of points i, j in the input space X. These neighbourhood relations are represented as a weighted graph G over the data points, with edges of weight $dX(i, j)$ between neighbouring points.
b. Compute shortest paths
   Initialize $d_G(i, j) = d_x(i, j)$ if i, j are linked by an edge; $d_G(i, j) = \infty$ otherwise. Then for each value of $k = 1, 2, \ldots, N$ in turn, replace all entries $d_G(i, j)$ by $\min\{d_G(i, j), d_G(i,k) + d_G(k, j)\}$. The matrix of final values $D_G = \{d_G(i, j)\}$ will contain the shortest path distances between all pairs of points in G. ISOMAP estimates the geodesic distances $dM(i, j)$ between all pairs of points on the manifold M by computing their shortest path distances $dG(i, j)$ in the graph G.
c. Construct d-dimensional embedding
   Let λp be the *p-th* eigenvalue (in decreasing order) of the matrix $t(D_G)$, and $v'p$ be the *i-th* component of the *p-th* eigenvector. Then set the *p-th* component of the d-dimensional coordinate vector $\boldsymbol{y}_i$ equal to $\sqrt{\lambda_p} v'_p$. Final step applies classical MDS to the matrix of graph distances DG 5 $\{dG(i, j)\}$, constructing an embedding of the data in a d-dimensional Euclidean space Y that best preserves the manifold's estimated intrinsic geometry. The coordinate vectors $\boldsymbol{y}_i$ for points in Y are chosen to minimize the cost function $E = \left\|\tau(D_{G)} - \tau(D_Y)\right\|L^2$, where D_Y denotes the matrix of Euclidean distance $\{d_Y(i,j) = \|y_i - y_j\|$ and $\|A\|L^2$ the matrix L^2 matrix norm $\sqrt{\sum_{i,j} A^2{}_{i,j}}$. The operator converts distances to inner products, which uniquely characterize the geometry of the data in a form that supports efficient optimization.

B. Kernel PCA

Kernel PCA is an extension of PCA [19], where PCA as a basis transformation to diagonalize an estimate of the covariance matrix of the data x_k, $k = 1,\ldots,\ell$ $x_k \epsilon R^N$, $\sum_{k=1}^{l} x_k = 0$, defined as $C = \frac{1}{\ell} \sum_{j=1}^{y} x_j x_j^{\mathrm{T}}$. The Kernel PCA algorithm proceeds as follows:

a. Set a kernel mapping $k(x_m, x_n)$.
b. Count $\boldsymbol{K}$ based on $\{x_n, (n = 1, \ldots, N)\}$.
c. Find eigenvalue of $\boldsymbol{K}$ to get λ_i and a_i
d. For each given data point $\boldsymbol{X}$, find principal components in the feature space: $\left(f(x) \,.\, \phi_i\right) = \sum_{n=1}^{N} a_k^{(i)} k(x, x_n)$

In this paper, Gaussian kernel applied $k(x, y) = exp\left\{ \frac{\left(-\|x - y\|^2\right)}{2\sigma^2} \right\}$

C. LLE

The LLE (Local Linear Embedded) algorithm based on simple geometric intuitions, where suppose the data consist of N real valued vectors $\vec{x}$ each of dimensionality, sampled from some smooth underlying manifold, the algorithm proposed [20]:

a. Compute the neighbours of each data point, $\overrightarrow{x_i}$
b. Compute the weight W_{ij} that best reconstruct each data point $\overrightarrow{X_i}$ from its neighbours, minimizing the cost in $\varepsilon(W) = \sum_i \left|\overrightarrow{x_i} - \sum_j W_{ij} \vec{X}_j\right|^2$ by constrained linear fits.
c. Compute the vectors $\vec{Y}_i$ best reconstructed by the weight W_{ij}, minimizing the quadratic form in $\Phi(\mathrm{Y}) = \sum_{\mathrm{i}} \left|\overrightarrow{\mathrm{Y}_{\mathrm{i}}} - \sum_{\mathrm{i}} \mathrm{W}_{\mathrm{ij}} \vec{\mathrm{Y}}_{\mathrm{j}}\right|^2$

D. MVU

Maximum Variance Unfolded (MVU) is algorithms for nonlinear dimensionality reduction [21] map high dimensional inputs $\{\overrightarrow{x}_i\}_1^n = 1$ to low dimensional outputs $\{\overrightarrow{y}_i\}_1^n = 1$, where $\vec{x}_i \epsilon \Re^r, \vec{y}_i \epsilon \Re^r and\ r \ll d$. The reduced dimensionality r chosen to be as small as possible, yet sufficiently large to guarantee that the outputs $\vec{y}_i \epsilon \Re^r$ provide a faithful representation of the input s$\vec{x}_i \epsilon \Re^r$.

E. PCA

Principal Component Analysis (PCA) is a dimension reduction technique that uses variance as a measure of interestingness and finds orthogonal vectors (principal components) in the feature space that accounts for the most variance in the data [22]. Principal component analysis is probably the oldest and best known of the techniques of multivariate analysis, first introduced by Pearson, and developed independently by Hotelling [23].

The advantages of PCA are identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. It is a powerful tool for analysing data by finding these patterns in the data. Then compress them by dimensions reduction without much loss of information [24]. Algorithm PCA [25] shown as follows:

a. Recover basis:

Calculate $XX^T = \sum_{i=1}^{t} x_i x_i{}^T$ and let U = eigenvectors of XX^T corresponding to the top d eigenvalues.

b. Encode training data:
$Y = U^T X$ where Y is a d x t matrix of encodings of the original data.

c. Reconstruct training data:
$\hat{X} = UY = UU^T X$

d. Encode test example:
$y = U^T > X$ where y is a d-dimensional encoding of x.

e. Reconstruct test example:
$\hat{x} = U_y = UU^T x$

3 Material and Method

This study is designed to find the most efficient dimension reduction technique. In order to achieve this objective, we propose a model for efficiency of the cluster performed by first reducing the dimensions of datasets. There are five dimension reduction techniques tested in the proposed model, namely ISOMAP, KernelPCA, LLE, MVU, and PCA.

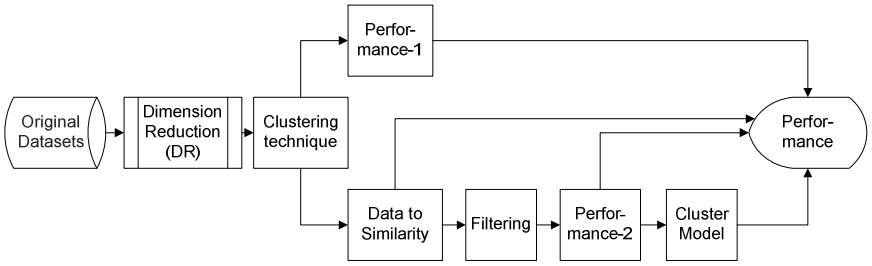

Fig.3. Proposed model compared based on dimension reduction and DBSCAN clustering.

Dimensions reduction result is processed into DBSCAN cluster technique. DBSCAN needs *ε (eps)* and the minimum number of points required to form a cluster (*minPts*) including mixed euclidean distance as distance measure. For the result of DBSCAN clustering using functional data to similarity, it calculates a similarity measure from the given data (attribute based), and another output of DBSCAN that is measured is *performance-1,* this simply provides the number of clusters as a value.

Result of *data to similarity* takes an exampleSet as input for filter examples and returns a new exampleSet including only the examples that fulfil a condition. By specifying an implementation of a condition, and a parameter string, arbitrary filters can be applied and directly derive a *performance-2* as measure from a specific data or statistics value, then process expectation maximum cluster with parameter *k*=2, *max runs*=5, *max optimization step*=100, *quality*=1.0E-10 and *install distribution*=*k-means* run.

4 Result

Testing of model performance was conducted on four datasets model; e-coli, iris, new machine cpu and thyroid. Dimension reduction used Isomap, Kernel PCA, LLE and MVU. Cluster technique used DBSCAN, Kernel K-Mean, Super Vector and Random Cluster. By using RapidMiner, we conducted the testing process without dimension reduction and clustering, and then compared with the results of clustering process using dimension reduction. Result of e-coli datasets process for processing time shown in Table 1a, and Table 1b for the performance of the cluster.

Table 1a. Processing time for e-coli datasets

Dimension reduction	Cluster Method			
	DBSCAN	Kernel K-Mean	Super Vector	Random Cluster
with ISOMAP	13	17	18	18
with Kernel PCA	14	24	15	14
with LLE	13	23	19	17
with MVU	13	18	15	15
with PCA	12	17	15	14
without dimension reduction	14	23	16	14

Table 1b. Performance of cluster for e-coli datasets

Dimension reduction	Cluster Method			
	DBSCAN	Kernel K-Mean	Super Vector	Random Cluster
with ISOMAP	99,4%	98,7%	99,4%	97,0%
with Kernel PCA	99,4%	98,7%	99,4%	97,1%
with LLE	99,4%	98,8%	99,4%	97,0%
with MVU	99,4%	98,7%	99,4%	97,0%
with PCA	99,4%	98,7%	99,4%	97,1%
without dimension reduction	99,4%	99,1%	99,4%	97,2%

Clustering process to iris datasets shown in Table 2a, for processing time and in Table 2b for the performance of the cluster.

Table 2a. Processing time for iris datasets

Dimension reduction	Cluster Method			
	DBSCAN	Kernel K-Mean	Super Vector	Random Cluster
with ISOMAP	11	6	6	6
with Kernel PCA	12	4	3	3
with LLE	11	10	7	7
with MVU	11	8	6	6
with PCA	10	5	4	4
without dimension reduction	11	8	7	7

Table 2b. Performance of cluster for iris datasets

Dimension reduction	Cluster Method			
	DBSCAN	Kernel K-Mean	Super Vector	Random Cluster
with ISOMAP	97,9%	93,5%	97,8%	91,2%
with Kernel PCA	98,7%	98,0%	98,7%	91,2%
with LLE	97,9%	95,6%	97,9%	91,2%
with MVU	97,9%	95,5%	97,8%	91,2%
with PCA	97,0%	98,0%	96,9%	93,9%
without dimension reduction	97,0%	98,0%	96,7%	93,9%

Machine cpu datasets consisting of 7 attributes and 209 samples clustered using the same method, and obtained the results shown in Table 3a, for processing time and Table 3b as a result of performance of the cluster.

Table 3a. Performance of cluster for machine cpu datasets

Dimension reduction	Cluster Method			
	DBSCAN	Kernel K-Mean	Super Vector	Random Cluster
with ISOMAP	11	3	4	5
with Kernel PCA	10	6	4	5
with LLE	8	4	5	5
with MVU	12	4	3	2
with PCA	11	7	9	7
without dimension reduction	13	15	22	19

Table 3b. Performance of cluster for machine cpu datasets

Dimension reduction	Cluster Method			
	DBSCAN	Kernel K-Mean	Super Vector	Random Cluster
with ISOMAP	98,6%	94,3%	33,3%	88,9%
with Kernel PCA	99,1%	66,7%	99,0%	95,4%
with LLE	97,2%	93,1%	97,2%	95,4%
with MVU	98,7%	99,4%	98,6%	88,9%
with PCA	40,0%	98,2%	0%	95,4%
without dimension reduction	99,5%	98,2%	99,5%	95,4%

Clustered result of new thyroid datasets shown in Table 4a for the processing time, and Table 4b for the performance of the cluster.

Table 4a. Performance of cluster for new thyroid datasets

Dimension reduction	Cluster Method			
	DBSCAN	Kernel K-Mean	Super Vector	Random Cluster
with ISOMAP	13	11	8	7
with Kernel PCA	17	7	9	9
with LLE	20	13	11	11
with MVU	17	13	12	9
with PCA	14	8	11	7
without dimension reduction	13	7	13	8

Table 4b. Performance of cluster for new thyroid datasets

Dimension reduction	Cluster Method			
	DBSCAN	Kernel K-Mean	Super Vector	Random Cluster
with ISOMAP	99,5%	96,9%	0%	95,5%
with Kernel PCA	99,1%	96,7%	99,1%	95,5%
with LLE	99,1%	98,9%	99,1%	95,5%
with MVU	98,7%	96,9%	0%	95,5%
with PCA	98,7%	96,7%	0%	95,5%
without dimension reduction	99,5%	98,3%	0%	95,6%

By implementing four different reduction techniques ISOMAP, KernelPCA, LLE, MVU, and PCA, and continuously applying the cluster method based on cluster density, We obtained results for the datasets of E.coli datasets. Some of the result We present at Fig. 4a-c. Fig. 4a is the result of the cluster with DBSCAN method that does not use a dimension reduction. Fig. 4b is the result of DBSCAN cluster method as well but first using dimension reduction. While Fig. 4c is the result of the cluster by using Super Vector and also use the dimension reduction.

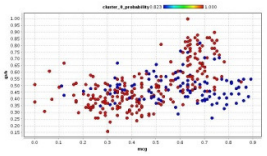

Fig. 4a. E-coli datasets based on DBSCAN without dimension reduction

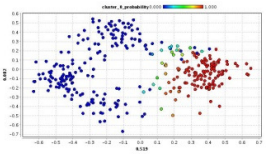

Fig. 4b. E-coli datasets based on DBSCAN and ISOMAP

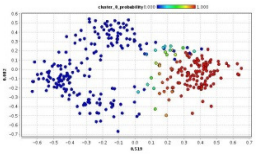

Fig. 4c. E-coli datasets based on Supervector and ISOMAP

For iris datasets consist of 4 attributes and 150 sample data, we implemented four different reduction techniques ISOMAP, KernelPCA, LLE, MVU, and PCA. We compared cluster result between process without dimension reduction and within dimension reduction. Some of the result present at Fig 5a-c. Fig. 5a is cluster result based on DBSCAN without dimension reduction. Fig. 5b is cluster result use DBSCAN within Kernel PCA as dimension reduction. This result similarly with Fig. 5c, cluster based on Random Cluster and Kernel PCA. Clustering process with dimension reduction create clearly different cluster (Fig. 5b and Fig. 5c).

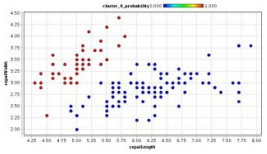

Fig. 5a. Iris datasets based on DBSCAN without dimension reduction

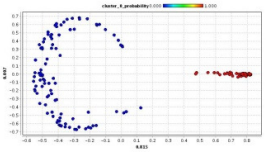

Fig. 5b. Iris datasets based on DBSCAN and Kernel PCA

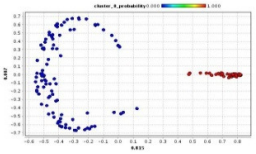

Fig. 5c. Iris datasets based on Random Cluster and Kernel PCA

The third was dataset tested is machine cpu. Some of the result we present at Fig 6a-c. In Fig. 6a shown cluster result based on DBSCAN without dimension reduction. Fig 6b. and Fig. 6c. was cluster result based on DBSCAN and Kernel K-Mean within using dimension reduction.

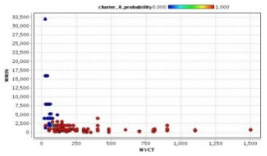

Fig. 6a. Machine cpu datasets based on DBSCAN without dimension reduction

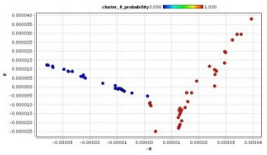

Fig. 6b. Machine cpu datasets based on DBSCAN and MVU

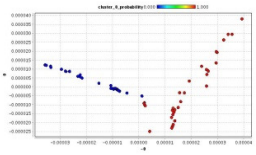

Fig. 6c. Machine cpu datasets based on Kernel K-Mean and MVU

Using same dimension reduction techniques, we clustered new thyroid datasets. We obtained results of DBSCAN without dimension reduction in Fig. 7a. While DBSCAN with dimension reduction using LLE has result in Fig. 7b. Cluster based Super Vector using LLE shown in Fig. 7c, we can see clustering process with dimension reduction create clearly different cluster (Fig. 7b. and Fig 7c.).

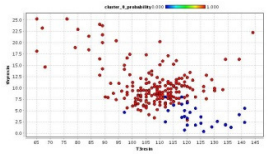

Fig 7a. Machine cpu datasets based on DBSCAN without dimension reduction

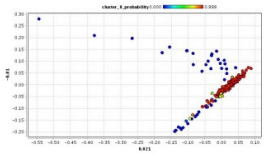

Fig 7b. Machine cpu datasets based on DBSCAN and LLE

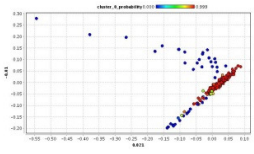

Fig 7c. Machine cpu datasets based on Super Vector and LLE

Each cluster process, especially ahead of determined value of $\varepsilon=1$, and the value $MinPts=5$, while the number of clusters ($k=2$) that will be produced was also determined before.

5 Discussion

Dimension reduction before clustering process is to obtain efficient processing time and increase accuracy of cluster performance. Based on results in previous section, dimension reduction can shorten processing time. Fig. 8a shows DBSCAN with PCA has lowest processing time.

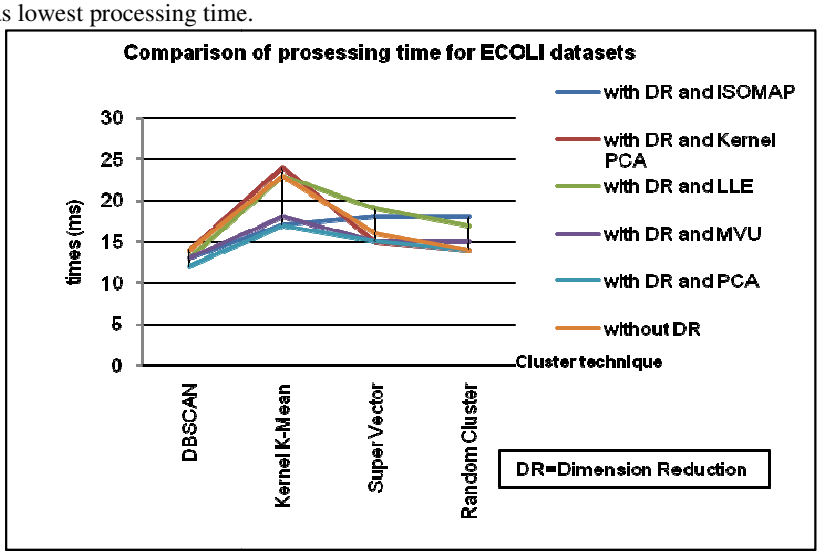

Fig. 8a. Performance of processing time for e-coli datasets using different dimension reduction technique and cluster technique

For iris datasets, we also found dimension reduction could shorten processing time. In Fig. 8b. Super Vector and Random Cluster within Kernel PCA has lowest processing time.

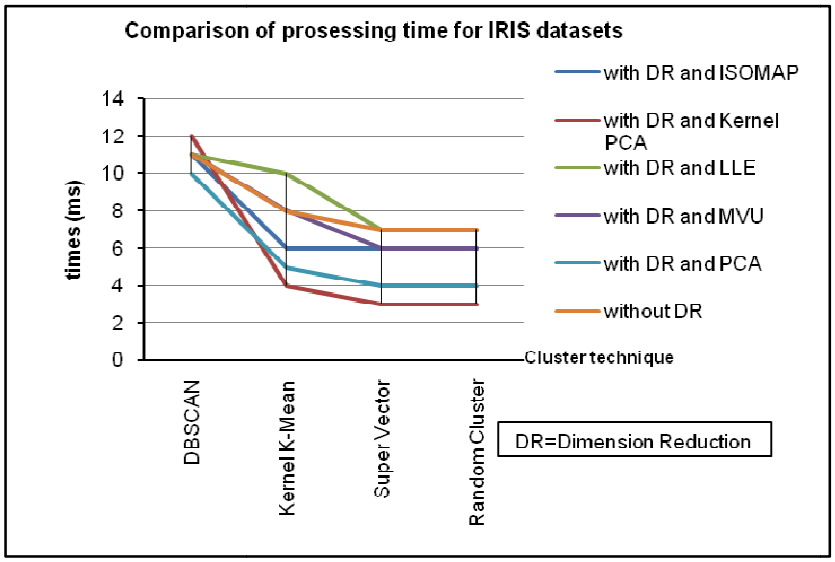

Fig. 8b. Performance of processing time for iris datasets using different dimension reduction technique and cluster technique

For machine cpu datasets, we found dimension reduction for Super Vector and Random Cluster within Kernel ISOMAP has lowest processing time (Fig. 8c).

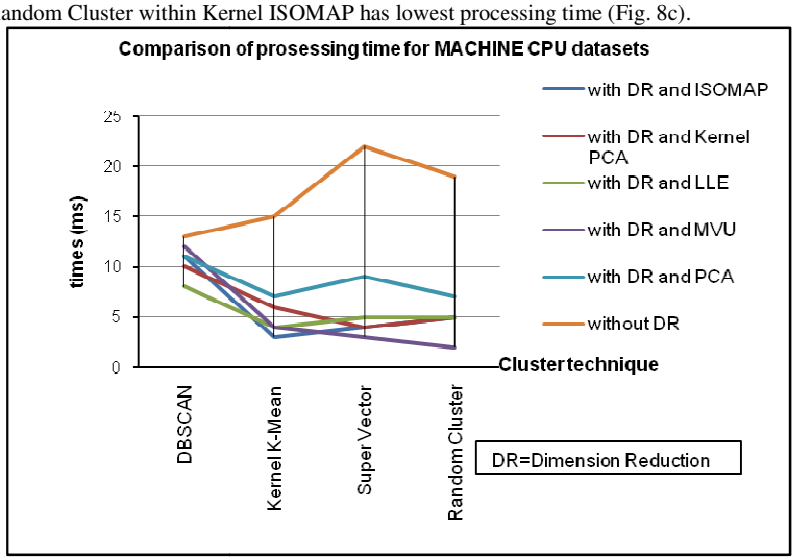

Fig. 8c. Performance of processing time for machine cpu dataset using different dimension reduction technique and cluster technique

For new thyroid datasets, we found dimension reduction for Kernel K-Mean within Kernel PCA and Random Cluster within Kernel ISOMAP has lowest processing time (Fig. 8d).

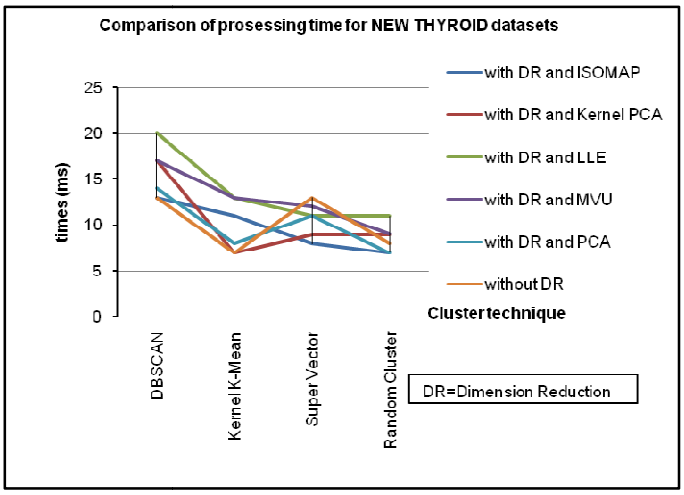

Fig. 8d. Performance of processing time for new thyroid datasets using different dimension reduction technique and cluster technique

Another evaluation for model implementation is comparison of cluster performance. In general dimension reduction increased cluster performance. For ecoli datasets we found Super Vector ISOMAP has highest cluster performance (Fig. 9a.).

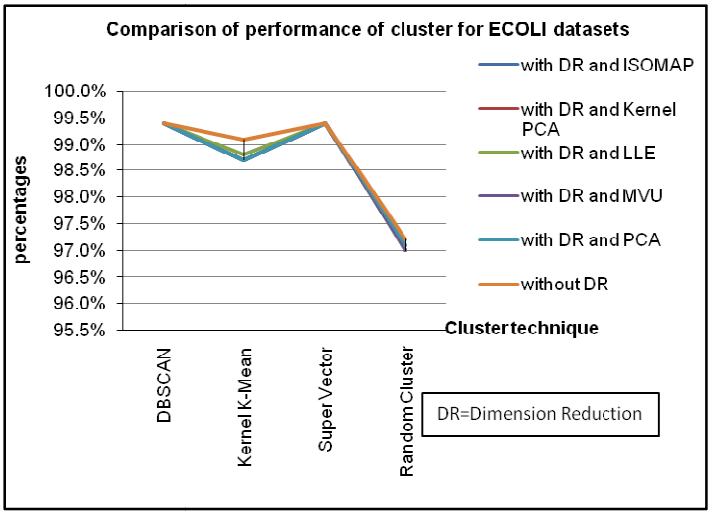

Fig. 9a. Performance of cluster for e-coli datasets using different dimension reduction technique

For iris dataset we found DBSCAN within Kernel PCA and Super Vector within Kernel PCA have highest cluster performance compared with cluster without dimension reduction (Fig. 9b.).

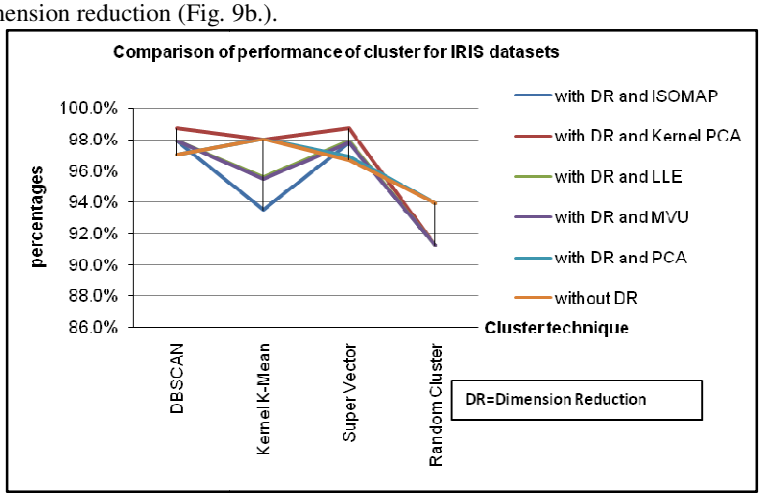

Fig. 9b. Performance of cluster for iris datasets using different dimension reduction technique

For machine cpu dataset in general cluster process without dimension reduction have highest cluster performance. Datasets, only Kernel K-Mean within PCA has cluster performance equal to cluster without dimension reduction (Fig. 9c.).

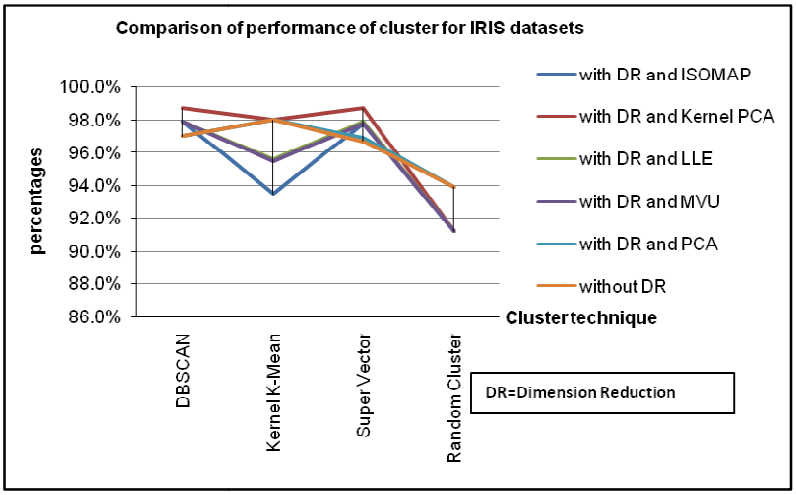

Fig. 9c. Performance of cluster for machine cpu datasets using different dimension reduction technique

For new thyroid dataset, we found Kernel K-Mean within LLE and Super Vector within LLE has highest cluster performance (Fig. 9d.).

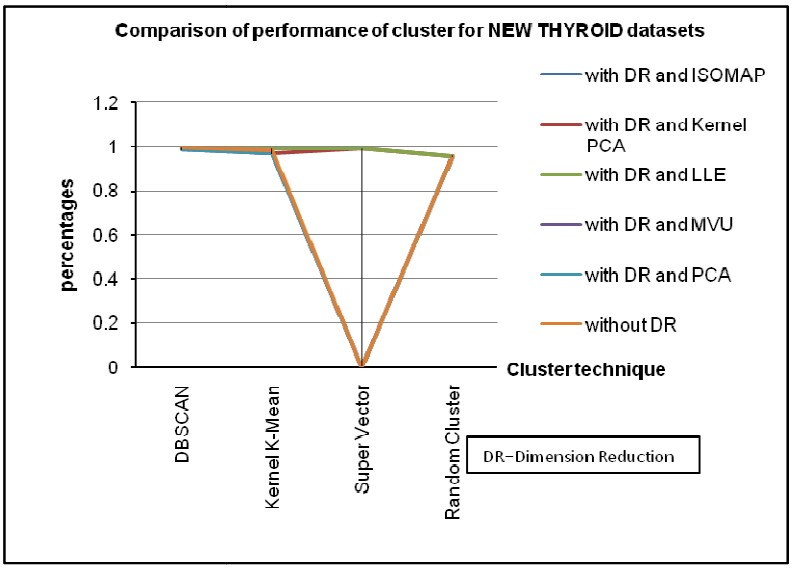

Fig. 9d. Performance of processing time for new thyroid datasets using different dimension reduction technique and cluster technique

6 Conclusion

The discussion above has shown that applying a dimension reduction technique will shorten the processing time.

Dimension reduction before clustering process is to obtain efficient processing time and increase accuracy of cluster performance. DBSCAN with PCA has lowest processing time for e-coli datasets. Super Vector and Random Cluster within Kernel PCA has lowest processing time for iris datasets. For machine cpu datasets, we found dimension reduction for Super Vector and Random Cluster within Kernel ISOMAP has lowest processing time. For new thyroid datasets, we found dimension reduction for Kernel K-Mean within Kernel PCA and Random Cluster within Kernel ISOMAP has lowest processing time.

In general, dimension reduction shows an increased cluster performance. For e-coli datasets, we found Super Vector ISOMAP has highest cluster performance. For iris datasets, we found DBSCAN within Kernel PCA and Super Vector within Kernel PCA have highest cluster performance compared with cluster without dimension reduction. For machine cpu dataset, in general cluster process without dimension

reduction have highest cluster performance. For new thyroid datasets, we found Kernel K-Mean within LLE and Super Vector within LLE show the highest cluster performance.

References

1. Maimon, Oded, Lior Rokach: Decomposition Methodology For Knowledge Discovery And Data Mining, World Scientific Publishing Co, Pte, Ltd, Danvers MA, p. 253-255 (2005)
2. Fodor, I.K: A Survey of Dimension Reduction Techniques. LLNL Technical Report, UCRL-ID-148494, p.1-18 (2002)
3. Chakrabarti, Kaushik, Sharad Mehrotra: Local Dimensionality Reduction : A New Approach To Indexing High Dimensional Space, Proceeding Of The 26th VLDB Conference, Cairo, Egypt, p.89-100 (2000)
4. Ding, Chris, Xiaofeng He, Hongyuan Zha, Horst Simon: Adaptive Dimension Reduction For Clustering High Dimensional Data, Lawrence Berkeley National Laboratory, p.1-8 (2002)
5. Globerson, Amir, Naftaly Tishby: Sufficient Dimensionality Reduction, Journal Of Machine Learning, p.1307-1331 (2003)
6. Jin, Longcun, Wanggen Wan, Yongliang Wu, Bin Cui, Xiaoqing Yu, Youyong Wu: A Robust High-Dimensional Data Reduction Method, The International Journal Of Virtual Reality 9(1), p.55-60 (2010)
7. Sembiring, Rahmat Widia, Jasni Mohamad Zain, Abdullah Embong: Clustering High Dimensional Data Using Subspace And Projected Clustering Algorithm, International Journal Of Computer Science & Information Technology (IJCSIT) Vol.2, No.4, p.162-170 (2010)
8. Sembiring, Rahmat Widia, Jasni Mohamad Zain: Cluster Evaluation Of Density Based Subspace Clustering, Journal Of Computing, Volume 2, Issue 11, p.14-19 (2010)
9. Nisbet, Robert, John Elder, Gary Miner: Statistical Analysis & Data Mining Application, Elsevier Inc, California, p.111-269 (2009)
10. Maimon, Oded, Lior Rokach: Data Mining And Knowledge Discovery Handbook, Springer Science+Business Media Inc, p.94-97 (2005)
11. Kumar, Ch. Aswani: Analysis Of Unsupervised Dimensionality Reduction Technique, ComSIS Vol. 6, No. 2, p. 218-227 (2009)
12. van der Maaten, L. J. P., E.O. Postma, H.J. van den Herik: Dimensionality Reduction: A Comparative Review. Published online: http://www.cs.unimaas.nl/l.vandermaaten/dr/dimensionreduction_draft.pdf, p.1-22 (2008)
13. Xu, Rui, Donald C. Wunsch II: Clustering, John Wiley & Sons, Inc, New Jersey, p. 237-239 (2009)
14. Larose, Daniel T: Data Mining Methods And Models, John Wiley & Sons Inc, New Jersey, p.1-15 (2006)
15. Wang, John: Encyclopedia Of Data Warehousing And Data Mining, Idea Group Reference, Hershey PA, p. 812 (2006)

16. Tenenbaum, Joshua, Vin De Silva, John C. Langford: A Global Geometric Framework For Nonlinear Dimensionality Reduction, Science, Vol. 290 No. 5500, p. 2319-2323 (2000)
17. Balasubramaniam, Mukund: The Isomap Algorithm And Topological Scaling, Science 295, p.7a (2002)
18. www.wikipedia.com
19. Schölkopf, Bernhard, Alexander Smola, Klaus Robert Muller: Non Linear Kernel Principal Component Analysis, Vision And Learning, Neural Computation 10, p.1299-1319 (1998)
20. Saul, Lawrence K: An Introduction To Locally Linear Embedding, AT&T Labs–Research, http://www.cs.nyu.edu/~roweis/lle/papers/lleintroa4.pdf, p.1-13 (2000)
21. Weinberger, Kilian Q, Lawrence K. Saul: An Introduction To Nonlinear Dimensionality Reduction By Maximum Variance Unfolding, AAAI'06 Proceedings of The 21st National Conference On Artificial Intelligence - Volume 2, p. 1683-1686 (2006)
22. Poncelet, Pascal, Maguelonne Teisseire, Florent Masseglia: Data Mining Patterns : New Methods And Application, Information Science Reference, Hershey PA, p. 120-121 (2008)
23. Jolliffe, I.T: Principal Component Analysis, Springer Verlag New York Inc. New York, p. 7-26 (2002)
24. Smith, Lindsay I: A Tutorial On Principal Component Analysis, http://www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf p.12-16 (2002)
25. Ghodsi, Ali: Dimensionality Reduction, A Short Tutorial, Technical Report 2006-14, Department of Statistics and Actuarial Science, University of Waterloo, p. 5-6 (2006)