ISSN:0975-9646
Sohel Ahamd Khan et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 8 (6) , 2017, 590-594

A Data Clustering Using Modified Principal Component Analysis with Genetic Algorithm

Sohel Ahamd Khan[1], Prof. Malti Nagle[2]
[1]Department of Computer Science, Surabhi College of Engg. & Technology, Bhopal, India
Head of Department of Computer Science, Surabhi College of Engg. & Technology, Bhopal, India

Abstract-Current development in both hardware and software has allowed large scale data acquisition. Typical statistical and data mining methods (e.g., clustering, regression, classification and frequent pattern mining) work with "static" data sets, meaning that the complete data set is available as a whole to perform all necessary computations. Digital world is growing as much faster beyond the thinking. Every digital process produces and uses lots of data on daily basis. We must need to store that data production into future uses format. So many digital equipment and sensor network produce the very precious data stream. Data steam consist variant parameters that are correlated with each other. In this paper, we proposed the method to cluster the unsupervised data stream to uncorrelated supervised cluster based on Principal component analysis. It mines dynamically changing sensor streams for states of system. It can be used for detecting the current state and as well as predicting the coming state of system. So, in this way, we can monitor the behaviour of the system so that any interesting events, such as anomalies or outliers can be found out.

Keywords: data stream, unsurprised, principal component analysis, clustering algorithm

INTRODUCTION

Data Streams are temporally ordered, fast changing, massive and potentially infinite. Unlike traditional datasets, data streams flow in and out of a computer system continuously and with varying update rates. It may be impossible to store an infinite data stream or to scan through it multiple times due to its tremendous volumes. Data Stream algorithms face many challenges, and have to satisfy constraints such as bounded memory, single-pass, real-time response & concept drift. Thus, the classification & clustering data streams with these constraints is a challenging problem.

Latest technology can generate huge quantity of sensor data continuously. Even though a single object like smart phone can generate continuous data stream in large amount from its sensor. So it's very challenging to define the cluster of continuous generated data stream because the definition of cluster may change time to time when data is even continuously changes. As an instance, the definition of object's may change when he or she walking, running or simple moving, for the condition person may get older that effect on motion sensor data.

Clustering of data stream is challenging because of limited memory and unbound less of data stream. Now a day's number of data clustering algorithm is present but these are not suitable for clustering of contextual sensor data stream.

In this paper we proposed MPCA-GA algorithm; the name "MPCA-GA" is attributed to modified principal component analysis of the distribution of data stream which are used to detect dynamically data stream and clustering it. Well known methods like k-means clustering, linear regression, decision tree induction and the APRIORI algorithm to find frequent itemsets scan the complete data set repeatedly to produce their results. However, in recent years more and more applications need to work with data which are not static, but are the result of a continuous data generating process. Data streams are ordered and potentially unbounded sequences of data points created by a typically non-stationary data generating process**.**

Nevertheless, dealing with huge amount of data poses a challenge for researchers, due to the limitations of current computational resources. For the last decade, we have seen an increasing interest in handling and managing these massive, unbounded sequences of data that are continuously generated at rapid speed over the period of time, the so-called data streams. More formally, a data stream S is a large sequence of data objectsX^1, X^2,…, X^N, that is, $S=\{X^i\}$ where i =1 to N, which is potentially unbounded ($N\longrightarrow \infty$). Each data object is described by an n-dimensional attribute vector$X^i = [x^i{}_1, x_1^i, x_2^i, \ldots, x_n^i]$, it is an attribute space omega that can be continuous, categorical, or mixed.

Data stream mining is very popular research area that has recently emerged to find knowledge from large amounts of continuously generated data. Such as from network flows, sensor data and web click stream. Examination and mining of such kind of data has been become very hot topic. Detecting hidden patterns in data stream give a great challenge for cluster analysis.

The main goal of clustering is to group the stream data and give some meaningful classes. Clustering problem needs to be defined carefully in this area. This is because a data stream should be viewed as an infinite process of continuously evolving data over time. For example, in network monitoring data, the TCP connection records of LAN (or WAN) network traffic, form a data stream. User network connection pattern often change gradually over time.

LITERATURE REVIEW

Clustering is a widely studied problem in the data mining literature. However, it is more difficult to adapt arbitrary clustering algorithms to data streams because of one-pass constraints on the data set.

Abdulhakim Qahtan, et al, Detecting changes in multidimensional data streams is an important and challenging task. In this paper they present the new frame work to detecting abrupt changes in multidimensional data streams. The framework is based on projecting data on selected principal components. On each projection, densities in reference and test windows are estimated and compared. Then a change-score value is calculated by one of the divergence metrics. By treating all selected PCs with equal importance, the maximum change-score among different PCs is considered as the final change-score [1]. Xiangliang Zhang, et al, Dealing with non-stationary distributions is a key issue for many application domains, e.g., sensor network or traffic data monitoring This method focuses on learning a generative model of a data stream, with some specific features: the generative model is expressed through a set of exemplars, i.e., actual data items as opposed to centroids in order to accommodate complex (e.g., relational) domains; the generative model is available at any time, for the sake of monitoring applications; the changes in the underlying distribution are detected through statistical hypothesis testing. to deal with this kind of problem they developed affinity propagation algorithm with the combination of statistical change detection. The proposed algorithm is known as STRAP [2].In this proposed algorithm STRAP aims at clustering data streams with evolving data distributions. Paul Voigtlaender, The DenStream algorithm [3] consists of an online part, which maintains a set of micro-clusters as a summarized representation of the data distribution, and an offline part, which generates the final clusters on demand. The DenStream algorithm uses the damped window model [4] to express the idea that recent data objects more accurately mirror the current data distribution than older ones. Philipp Kranen, They propose a parameter-free stream clustering algorithm ClusTree [5] that is capable of processing the stream in a single pass, with limited memory usage. It always maintains an up-to-date cluster model and reports concept drift, novelty, and outliers. Moreover, their approach makes no apriori assumption on the size of the clustering model, but dynamically self-adapts. For handling of varying time allowances, we propose an anytime clustering approach. Anytime algorithms are capable of delivering a result at any given point in time, and of using more time if it is available to refine the result. For clustering, this means that our algorithm is capable of processing even very fast streams. Charu C. Aggarwal, et al, The CluStream methodology may be a methodology of clustering data streams, supported the conception of microclusters [8]. Microclusters are data structures that summarize a collection of instances from the stream, and consists of a collection of statistics that are simply updated and permit quick analysis. D-Stream [7] is a framework to clusterstream data that is work on a density-based approach. The algorithmic uses an online part that maps every input file record into a grid and an offline part that computes the grid density and clusters the grids supported the density. The algorithmic rule adopts a density decaying technique to capture the dynamic changes of an information stream. Exploiting the knotty relationships between the decay issue, data density and cluster structure, our algorithmic rule will expeditiously and effectively generate and change the clusters in real time.

PROPOSED WORK

Here we tend to propose MPCA-GA, a unique data-stream clustering algorithm for dynamically detecting and managing sequential temporal contexts. MPCA-GA takes into account the properties of sensor attached data streams in order to accurately conclude the present concept, and dynamically detect new contexts as they occur. Moreover, the algorithm is proficient of detecting point anomalies and can operate with high velocity data streams.

The Context Model

Here PCA captures the relationship of correlation between the dimensions of a collection of observation kept in $m \times n$ matrix X, where m is the number of observations and n is the number of attributes in the stream. When we apply PCA on X it gives two $n \times n$ matrices; one is the diagonal matrix V which consists of Eigen values and the other is an orthonormal matrix P which consists of Eigen-vectors (a.k.a. principal components). The Eigen vectors $\overrightarrow{p_1}$, $\overrightarrow{p_2}$, ..., $\overrightarrow{p_n}$ oriented according to the correlation of X and from a basis in $\boldsymbol{R}^n$ which is centered on X. The Eigen values of V are arranged from highest to lowest variance and their respective Eigen vectors which are stored in P are ordered accordingly. In other words, $\overrightarrow{p_1}$ is the direction of highest variance in the data (with σ^{12}) from the mean of the collection X.

Here we define the contribution of a particular component $\overrightarrow{p_i}$ as the percent of total variance it describe for the collection X.

$$cont_x = \frac{\sigma_i^2}{\sum_{j=1}^{n} \sigma_j^2}$$

By collecting only these PCs, we effectively compile the distribution and focus on our future calculation on the dimensions of interest.

Merging Model

There are lots of cases when we make changes in MPCA-GA parameters and by doing this we will create several models for the memory space of the main system. In worst situation (k_{c_i}=n) the memory required for matrix Mi is (mn) and for the context model c_i it is (n^2) for matrix A_i. So total memory space needed for MPCA-GA is ($|C|(n^2 + mn)$).

To overcome the memory limit drawback, we tend to propose that if we detect a new context and memory limit is reached, then we will merge two models into one new model. For selecting which model we have to merge,

$$merg(c_i, c_j) = c_l$$

Where model c_i and c_j is merged into a new context model c_l. We are accomplishing this in two ways. One method is to equally interleave the primary m observations of M_i and M_j into new M_l.

Similarity Scores

For calculating the similarity score, we use similar method that SIMCA used; therefore we use Mahalanobis distance to retain only the top k PCs of that distribution for finding point's statistical similarity to a distribution. For producing this score we first do zero-meaning at the point to that distribution, after that it is transformed into that distribution, then we take top k PCs and transforming it onto the distributions, and then by computing the magnitude of resulting point's. Formally, for distribution of ci the similarity of point $\vec{x}$ is

$$d_{c_i}(\vec{x}) = \| (\vec{x} - \mu_i)A_i \|$$

From the ellipsoid shapes extended from distribution and according to their variance in every direction, we can see the necessity of performing Mahalanobis distance.

Detection of New Context

A main functionality of the MPCA-GA algorithm is to detect a new context which is unseen previously. From Definition 4, contexts are implicit to have static distribution, but they may have changed progressively over the period of time as it is subjected to concept drift. Therefore, S that is generated by distribution of data, not fits in any contexts present in C for a constant t_{min} time ticks. By seeing this we can say that these t_{min} observations comprise a new context, and modeled for future use. A new context is like finding a new model in our dataset which is unseen we can understand this by example in KDD dataset we seen some unseen attacks.

To follow these activities, a new concept is introduced called "drift buffer". Let the name of drift buffer be D and it has a length of t_{min}. When D is filled without disturbance (i.e. should $D = \{\vec{x}_{\text{t-tmin}}, \ldots\ldots, \vec{x}_{\text{t}}\}$ then the contents present in D is emptied for creating a new context model, and it is set as current context c_{t}. In case of fractional drift (i.e. D will not completely fill, yet $\vec{x}_{\text{t}}$ fits some context in $\boldsymbol{C}$) we can say that S is experience a wider limit of c_{t}, consequently we empty D in c_{t}.

Algorithm: MPCA-GA$\{S\}$

Input parameters $\{\varphi_{,\text{min}}, m, \rho\}$

Anytime Outputs: $\{c_{\text{t}}, d_c(\overrightarrow{x_t})\}$

1. $\boldsymbol{C} \leftarrow (S, t_{\text{min}}, m, \rho)$
2. $c_{\text{t}} \leftarrow c_1$
3. $D \leftarrow \emptyset$
4. *loop*
   - 4.1. $\overrightarrow{x_c} \leftarrow$ next(S)
   - 4.2 $scores \leftarrow d_c(\overrightarrow{x_c})$
   - 4.3. $i \leftarrow$ Indes_Of_Min($scores$)
   - 4.4. *if* scores(i)< φ
     - 4.4.1. $Upd_$Model(c_{i},$Dump(D)$)
     - 4.4.2. $Upd_$Model(c_{i},$\overrightarrow{x_c}$)
     - 4.4.3. $c_{\text{t}} \leftarrow c_{\text{i}}$
   - 4.5. Else
     - 4.5.1. insert($\overrightarrow{x_c}$, D)
     - 4.5.2. *if* $length(D) == tmin$
     - 4.5.3. $c_{\text{t}} \leftarrow c_{\text{i}}$
5. *end loop*

RESULT AND ANALYSIS

Dataset and Test Performance

The proposed method implements in MATLAB R2013a and tested with very reputed data set from UCI machine learning research center. In the research work, I have measured CR (Correct Rate), ER (Error Rate), PPV (Positive Predictive Value), NPV (Negative Predictive Value), PL (Positive Likelihood) and NL (Negative Likelihood) of Clustering. To evaluate these performance parameters, it has been used three datasets from UCI machine learning repository namely glass dataset, banana dataset and forest fire dataset. Out of these three dataset, one is small dataset namely glass dataset and other one is large dataset namely as banana dataset. Here the comparisons result tested on the basis of vary chunk size and measure the various result parameters shown in the comparisons tables. Here the performance of the proposed work is better as compared to the existing technique. The proposed work takes less time as compared to the existing technique and the proposed work reduced error rate as compare to the existing method. Following are the result comparisons of both methods:

Comparison table for the PCA and MPCA-GA method on the basis of given different chunk size and find the value of CR, ER, PPV, NPV, PL and NL.

COMPARISON TABLE

CHUNK SIZE	PCA						MPCA-GA					
	CR	ER	PPV	NPV	PL	NL	CR	ER	PPV	NPV	PL	NL
80	51.28	48.72	0.221	1	1.63	0	82.05	17.94	1	0.82	1.77	0.8
90	56.8	43.18	0.406	1	1.56	0	70.45	29.54	0.5	0.73	2.38	0.85
100	63.26	36.73	0.41	1	1.22	0	65.31	34.69	0.57	0.67	2.29	0.86
110	53.7	46.3	0.47	1	1.24	0	55.56	44.44	0.45	0.58	1.12	0.97

Table 1: Show that the calculated result of feature selection on different chunk size on the basis of two methods PCA and MPCA-GA for glass data set.

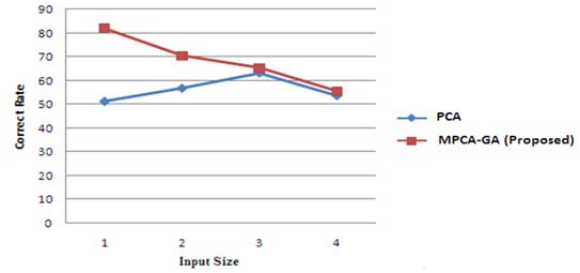

Figure 1: shows that the comparison graph correct rate of both methods PCA and MPCA-GA for glass data set

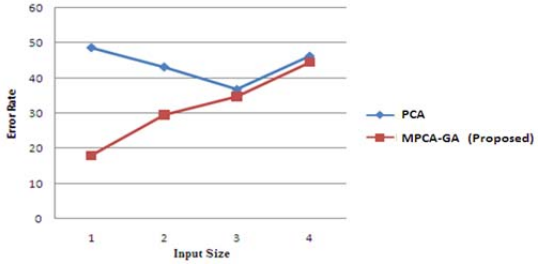

Figure 2: Shows that the comparisons graph between error rate of both methods PCA and MPCA-GA for glass data set.

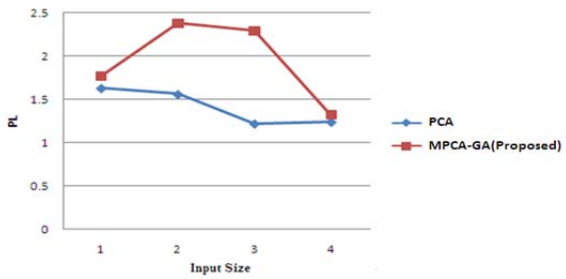

Figure 3: Shows that the comparisons graph between PL of both methods PCA and MPCA-GA for glass data set

Comparison table for the PCA and MPCA-GA method on the basis of given different chunk size and find the value of CR, ER, PPV, NPV, PL and NL.

COMPARISON TABLE

CHUNK SIZE	PCA						MPCA-GA					
	CR	ER	PPV	NPV	PL	NL	CR	ER	PPV	NPV	PL	NL
200	46.46	53.53	0.428	0.467	0.68	1.03	55.56	44.4	0.57	0.53	1.19	0.78
400	56.28	43.72	0.56	0.5	1.01	0.78	59.79	40.2	0.69	0.53	1.8	0.69
600	55.85	44.15	0.56	0	1	0	58.53	41.47	0.67	0.52	1.6	0.72
800	45.36	54.63	0.565	0.45	1.05	1	49.87	50.12	0.54	0.43	0.96	1.06

Table 2: Show that the calculated result of feature selection on different chunk size on the basis of two methods PCA and MPCA-GA for banana data set.

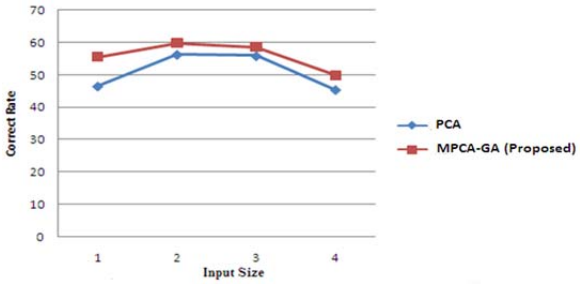

Figure 4: Shows that the comparisons graph between Correct Rate of both methods PCA and MPCA-GA for banana data set

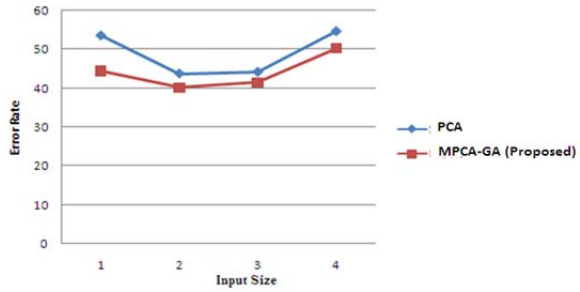

Figure 5: Shows that the comparisons graph between Error Rate of both methods PCA and MPCA-GA for banana data set

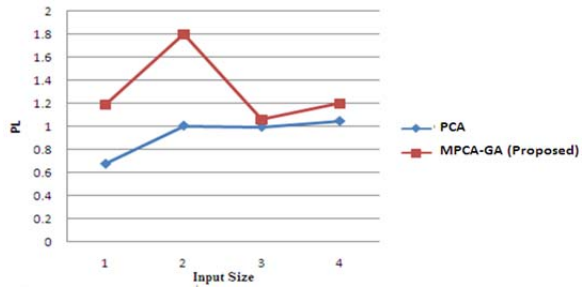

Figure 6: Shows that the comparisons graph between PL of both methods PCA and MPCA-GA for banana data set

CONCLUSION

Most techniques assume that the true label of a data point can be accessed as soon as it has been classified by the clustering model. Thus, according to their postulation, the existing model can be updated without delay using the labeled instance. In reality, one would not be so lucky in obtaining the label of a data instance immediately, since manual labeling of data is time consuming and costly. We claim two major contributions in novel class detection for data streams. First, we propose a dynamic selection of boundary for outlier detection by allowing a slack space outer the decision boundary. This space is restricted by a threshold, and the threshold is modified all the time to reduce the risk of false alarms and missed novel classes. Modified Principal Component Analysis is very efficient data mining tool for data clustering. Data clustering is challenging task in the field of clustering. Evaluation of new feature creates a problem in feature selection during the clustering process of modified Principal Component Analysis.

SCOPE OF FUTURE WORK

The proposed method Modified Principal Component Analysis solved the problem of feature evaluation and concept evaluation. The controlled feature evaluation process increases the value of correct rate and reduces the error rate. The genetic algorithm optimization cluster faced a problem of right number of cluster, in future used self optimal clustering technique along with other optimization technique is as particle of swarm optimization, continuous orthogonal ant colony optimization etc.

REFERENCE:

[1] Author, Prof. Malti Nagle, A PCA-based Clustering Algorithm for State Recognition in Sensor Data Streams, IJIRCCE, ISSN(Online): 2320-9801 ISSN (Print): 2320-9798 Vol. 5, Issue 3, March 2017

[2] Charu C. Aggarwal, "DATA STREAMS MODELS AND ALGORITHMS" IBM T. J. Watson Research Center, Yorktown Heights, NY 10598ISBN: 978-0-387-28759-1 (Print) 978-0-387-47534-9 (Online), Volume 31 2007.

[3] X Zhang, C Furtlehnery, C Germain-Renaudy, MichleSebage "Data Stream Clustering with Affinity Propagation" IEEE Transactions on Knowledge and Data Engineering, Institute of Electrical and Electronics Engineers, 2014, 26 (7).

[4] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, XiaoweiXu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", KDD-96 Proceedings, AAAI, 1996.

[5] Paul Voigtlaender, "DenStream algorithm for clustring", Rheinisch-WestfälischeTechnischeHochschule Aachen, January 22, 2013.

[6] T Dasu, S Krishnan, S Venkatasubramanian, "An Information-Theoretic Approach to Detecting Changes in Multi-Dimensional

Data Streams", In Proc. Symp. on the Interface of Statistics, Computing Science, and Applications, 2006.
[7] Jonathan Silva, "Data Stream Clustering: A survey", ACM Computing Survey, 2013.
[8] Yixin Chen, "Density-based clustering for real-time stream data", KDD '07 Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data miningPages 133-142 San Jose, California, USA — August 12 - 15, 2007.
[9] Albert Bifet, et al.: "Learning from Time-Changing Data with Adaptive Windowing", 6th Framework Program of EU through the integrated project DELIS (#001907), by the EU PASCAL Network of Excellence, IST-2002-506778, and by the DGICYT MOISES-BAR project, TIN2005-08832-C03-03.

AUTHOR

Sohel A Khan has completed his B.E. in Information Technology in 2011 from K. D. K College of Engineering, Nagpur (MAH), & Pursuing M.Tech from RGPV University. His main research interest is Machine Learning and Big Data Analyst.

Prof. Malti Nagle received the B.E. in Information Technology from Samrat Ashok Technological Institute VIDISHA (M.P.) in 2006 and M.Tech in Computer Science & Engineering from Jaypee University of Information Technology (U.P.) in 2009. She is a Professor of Computer Science & Engineering with the Surabhi College of Engineering BHOPAL, M.P. Prior to that, she led the IT Trainer at Smritinet.com, BHOPAL, M.P. Her main research interests are Network security and ADHOC Network in which she has published more than 15 papers. Prof Malti was an IEEE review committee member at SoftCOM 2014.She has been in education profession from 7 years. She worked in various university as A.P.