ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

A PCA-based Clustering Algorithm for State Recognition in Sensor Data Streams

Sohel A. Khan, Prof. Malti Nagle

M.Tech Student, Department of CSE, Surabhi College of Engg. & Technology, Bhopal (M.P.), India

Head of Department, Department of CSE, Surabhi College of Engg. & Technology, Bhopal (M.P.), India

ABSTRACT: 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 mothed 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 behavior 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

I. 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 its 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 a 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 days number of data clustering algorithm is present but these are not suitable for clustering of contextual sensor data stream.

In this paper we proposed pcaStream algorithm; the name "pcaStream" is attributed to principal component of the distribution of data stream which are used to detect dynamically data stream and clustering it.

II. LITERATURE REVIEW

1. Data Stream Clustering: A survey

The basic nature of stream knowledge needs the development of algorithms capable of acting quick and progressive process of information objects, fitly addressing time and memory limitations. This article [6] given survey ofknowledge streamcluster algorithms, providing a detail discussion of the most style parts of progressive algorithms like CluStream, D-Stream, Den-Stream, DBSCAN etc.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

2. A Micro-clusteringApproach

Clustering of data is defined as grouping of data on the basis of dissimilarity of object with use of some distance measure and objective functionality. The basic algorithm to micro clustering is derives from K means algorithm. To achieve goal this approach uses CluStream concept and pyramidal time window. The uses of a pyramidal time window guarantees that the crucial statistics of growing data streams can be captured without forgoing the underlying space- and time efficiency of the stream clustering procedure.

3. STRAP algorithm

Algorithm STRAP objectives at clustering data streams with growing data distributions. STRAP opposes the arriving items to the recent AP model, storing the outliers in a pool and monitoring the ratio of outliers using the Page-Hinkley change point detection test. Upon activating the PH test, the clustering model is recreated from the current one and the pool using WAP.

4. DenStream Algorithm

DenStream, a set of initial micro-clusters and a set of current micro-clusters is maintained. The algorithm is initialized with the DBSCAN algorithm applied to the initial InitN data points, which generates initial init-micro-clusters. Later the initialization phase for each new data entry, a merging algorithm is used to maintain the current-micro-clusters. Additionally, a pruning algorithm is executed periodically to remove outliers when a clustering request is received, the offline part of the algorithm generates the finalclusters using a variant of the DBSCAN algorithm, which uses the maintained-micro-clusters as virtual points.

5. DBSCAN Algorithm

DBSCAN lives on a density-based notion of clusters. It requires only one input parameter and supports the user in determining an appropriate value for it. DBSCAN performed a performance evaluation on synthetic data and on real data. DBSCAN is finding non-linear shapes structure based on the density. DBSCAN is most extensively used density based algorithm

6. D - Stream Algorithm

D-Stream [7] is a framework to cluster stream 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.

III. PROPOSED WORK

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 $\boldsymbol{R}^{\mathrm{n}}$ which is centred 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.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

Here we define the contribution of a particular component $\vec{p_i}$ 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.

Let target percent of variance to be retained is ρ. We define $k \in N$ to be the highest PCs in which cumulative sum of contribution of these PCs surpasses ρ. Stated

otherwise, argument $min\{\sum_{i=1}^{k} contx_x(\vec{p_i}) \geq \rho\}$. Here the k associated with context c_i denoted as k_{c_i}.

We model the i[th] discovered context as the tuple $c_i = (M_i, \mu_i, A_i)$, Where Mi is a $m \times n$ for the last m observations assigned to c_i, μ_i is the mean of observation in Mi, and lastly

$A_i = \left[\frac{\vec{p_1}}{\sigma_1}, \frac{\vec{p_2}}{\sigma_2}, \ldots . \frac{\vec{p_{k_{c_i}}}}{\sigma_{k_{c_i}}}\right]$ is a $n \times k$ matrix.

Basically A_i is a truncated version of P with its Eigen-vectors normalized to their standard deviation. For getting A_i first we perform PCA on M_i, so we can get P_i and V_i, and by then calculating $A_i = Q_i \Lambda_i$ where Qi is the column wise truncation of Pi so that we can get first k_{c_i}. columns and Λ_i is a diagonal matrix of the top k_{c_i} largest SDc (which are obtained from Vi).

When a new point is added in cluster c_i, matrix M_i acts as a windowed memory for c_i and it will discard the m^{th} oldest observation. Windowing over a stream is an implied methodology for managing concept drift.

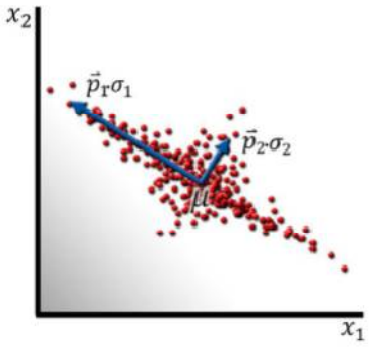


Fig.1 Graphical representation of principal components($\vec{p_1}, \vec{p_2}$) scaled to respective standard deviation (σ_1 , σ_2) and centered on the distribution's mean(μ).

Basically Ai is a truncated version of P with its Eigen-vectors normalized to their standard deviation. For getting Ai first we perform PCA on Mi, so we can get Pi and Vi, and by then calculating $Ai = Qi\Lambda i$ where Qi is the

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

column wise truncation of Pi so that we can get first kci columns, and Λi is a diagonal matrix of the top kci largest SDc (which are obtained from Vi).

When a new point is added in cluster ci, matrix Mi acts as a windowed memory for ci and it will discard the mth oldest observation. Windowing over a stream is an implied methodology for managing concept drift.

Merging Model

There are lots of cases when we make changes in pcaStream 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 pcaStream 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. By doing this we make sure that at minimum half of each context's information are going to be preserved.

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. To give strength to our concept further, fig illustrates the instance where for point$\vec{x}$ two contexts (c_i andc_j) are compared in similarity. Their similarity scores are different when we take the top k_{c_i} and k_{c_j} normalized PCs of each individual context, even though if we use Euclidean distance between point $\vec{x}$ and μ_i and μjthey are equivalent. Moreover, depending on ρ and the relationship of the distribution it is possible that the k_{c_i}>k_{c_j}.

Let the similarity threshold is define as $\varphi \in [0,\infty]$. We say that if context ci is $d_{c_i}(\vec{x}) > \varphi$ than a point $\vec{x}$ is not similar. Let score vector of $\vec{x}$ of each model present in C is defined as $d_c(\vec{x}) = [d_{c_1}(\vec{x}), d_{c_2}(\vec{x}), \ldots\ldots, d_{c_{|c|}}(\vec{x})]$. If $d_{c_i}(\vec{x}) > \varphi$ than we can say that $\vec{x}$ does not fit any known context. Moreover, we say that $\hat{x}$ is most similar to context ci if the smallest element in $d_c(\vec{x})$ is $d_{c_i}(\vec{x})$.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

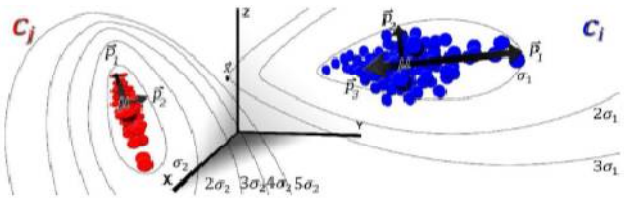
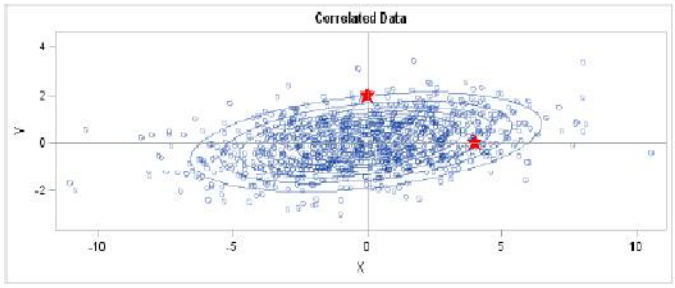


Fig. 2. An illustration of how the ellipsoids of the Mahalanobis distance affect the score of a point ($\vec{x}$)calculated in the perspective of the contexts $\boldsymbol{c}_i$and $\boldsymbol{c}_j$. Here $d_{c_i}(\vec{x}) < d_{c_j}(\vec{x})$.

Why we use Mahalanobis distanceinstead of Euclidean distance we can understand this by the help of this graph



The following graph shows bivariate normal data that is overlaid with prediction ellipses. The ellipses in the graph are the 10% (innermost), 20%, …. and 90% (outermost) prediction ellipses for the bivariate normal distribution that generates the data. The prediction ellipses are contours of the bivariate normal density function. The probability density is high for the ellipses near the origin, such as the 10% prediction ellipse.

In this graph, two observations are displayed by using red stars as markers. The first observation is at the coordinates (4,0), whereas the second is at (0,2). Now the question is which marker is closer to the origin?

Detection of New Context

A main functionality of the pcastream 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 19

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 $D = \{\vec{x}_{\text{t-tmin}},.....,\vec{x}_{\text{t}}\}$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

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

not completely fill, yet $\vec{x}_t$ 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.

IV. CORE PCASTREAM ALGORITHM

Online Algorithm 1: pcastream$\{S\}$
Input parameters $\{\varphi,n,m,\rho\}$

Anytime Outputs: $\{ct,(\vec{xt})\}$

1. $\boldsymbol{C} \leftarrow i(S,tmin,m,\rho)$

2. $ct \leftarrow c1$

3. $D \leftarrow \emptyset$

4. $loop$
4.1. $\vec{xc} \leftarrow n(S)$
$scores \leftarrow dc(\vec{xc})$
4.2. $i \leftarrow indexOf(scores)$

4.3. $if\ sco(i) < \varphi$
4.3.1. $UpdateMo(ci,Dump(D))$

4.3.2. $UpdateMo(ci,\vec{xc})$

4.3.3. $ct \leftarrow ci$
4.4. Else
4.4.1. $ins(\vec{xc},D)$

4.4.2. $if\ length(D) == tmin$

4.4.3. $ct \leftarrow ci$
5. $end\ loop$

V. RESULT AND ANALYSIS

Dataset and Test Performance

The pcastream algorithm and other data stream algorithms are run in R (a software environment for statistical computing). Main packages in R which are used for data stream are stream and streamMOA. The streamMOA package is basically an interface it provide an interface to algorithm implemented for the MOA (Massive Online Analysis) framework for data stream mining.

We used KDD'99 network intrusion dataset. KDD training dataset consist of approximately 4,900,000 single connection vectors, each of which contain 41 features and is labelled as either normal or an attack, with exactly one specific attack type. The attacks fall in one of the following four categories:

1) Denial of Service Attack (DoS)
2) User to Root Attack (U2R)
3) Remote to Local Attack (R2L)

4) Probing Attack

Main problem in original KDD dataset is the huge number of redundant records. Analyzing KDD train and test sets, we found that about 78% and 75% of the records are duplicated in the train and test set, respectively. This large amount of redundant data in the train set will cause learning algorithm to be biased towards the more frequent records, and thus prevent it from learning unfrequented records which are usually more harmful to networks such as

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 3, March 2017

U2R attacks. The existence of these repeated records in the test set, on the other hand, will cause the evaluation results to be biased by the method which have better detection rate on frequent records.

So we can use 10% of original dataset that is approximately 494,020 single connection vectors each of which contain 38 features. A smaller version 10% training dataset is also provided for memory constrained machine learning methods.

The major objectives performed by detecting network intrusion are stated as recognizing rare attack types such as U2R and R2L. This detects that the training dataset consist of 494,019 records, among which 97,277 (19.69%) were 'normal', 391,458 (79.24%) DOS, 4,107 (0.83%) Probe, 1,126 (0.23%) R2L and 52 (0.01%) U2R attacks.

VI. CONCLUSION

We studied pcaStream algorithm, a stream clustering algorithm that dynamically detects the temporal contexts in sensor data stream. In addition, our pcaStream algorithm accounts for gradual concept drifts and clusters which overlap with each other in geometric space. We showed a mechanism for dealing with high velocity stream.

REFERENCES

[1] 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.
[2] X Zhang, C Furtlehnery, C Germain-Renaudy, Michle Sebage "Data Stream Clustering with Affinity Propagation" IEEE Transactions on Knowledge and Data Engineering, Institute of Electrical and Electronics Engineers, 2014, 26 (7).
[3] Martin Ester, Hans-Peter Kriegel, Jiirg Sander, Xiaowei Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise", KDD-96 Proceedings, AAAI, 1996.
[4] Paul Voigtlaender, "DenStream algorithm for clustring", Rheinisch-Westfälische Technische Hochschule Aachen, January 22, 2013.
[5] 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.
[6] Jonathan Silva, "Data Stream Clustering: A survey", ACM Computing Survey, 2013.
[7] 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 mining Pages 133-142 San Jose, California, USA — August 12 - 15, 2007
[8]. Amineh Amini,The Ying Wah: Density Micro Clustering Algorithm on Data Stream. IMECS 2011 Hong Kong.