**CHANGE DETECTION IN ONLINE MULTIVARIATE SENSOR DATA**

B.Tech Term Project Report

By

**BIMAL KUMAR SAHOO**

**17IM30032**

Under the supervision of

**MAMATA JENAMANI**



**Department of Industrial & Systems Engineering**

**Indian Institute of Technology Kharagpur**

**Kharagpur-721302, India**

**2020**

**ABSTRACT**

This report is based on a change detection algorithm used for detecting changes in fruits, here mangoes during transportation or during the storage in the inventory.

The data is received from nodes connected into Wireless Sensor Networks (WSNs).

The data received is multivariate data from different types of nodes like humidity, temperature, ammonium emission etc and from nodes in different locations. This paper shows a possible implementation of a non-parametric method to detect changes in live multivariate data from the sensor nodes. The method used is a semi parametric log likelihood(SPLL) method with clustering algorithms which detects changes in probability distribution of the data recorded. The SPLL method is modified to be used for online streaming data by adding sliding window methods. In this paper, we have compared two methods, one where the algorithm compares the online data with a fixed baseline window and another with a sliding base window. This method compares the SPLL method with KL divergence method and with Hotelling’s t2-test. Method analysis is done using ROC curve and AUC to select the best algorithm for this problem statement. Then In-control run length and Out-of-control run length metrics are used to select between adaptive baseline sliding window or fixed baseline sliding window. This report also explains possible future implementation of a protective mechanism for reducing the number of false alarms in successive iterations.

**INTRODUCTION**

Change detection in streaming data is the process of identifying unexpected deviations from normal behaviors. Change detection has many applications like cloud server monitoring, environment changes network anomalies detection, etc. Many change detection methods are developed over years, first being basic mean deviation. There are lots of classes of change detection algorithms, like sequential algorithm (CUSUM, Page-Hinkley, etc), control charts (Geometric moving average chart, Exponentially weighted moving average etc) and monitoring distributions (SPLL, Log Likelihood KL divergence, log likelihood Hotelling’s t2 test). Change detection algorithms are also used to detect changes in the probability distribution of stochastic (or time series) data. Change detection is widely used in fields where we have a continuous stream of time series data or an offline data store to record for changes occuring in previous data.

On the basis of data collection, there are two categories of change detection algorithms:

Offline data: The entire dataset is available before applying the model. In this type of methods, different clustering techniques are used to cluster the data set as well as the base data set. Then a change detection algorithm is run to detect the changes occuring in particular data points.

Online data: The change detection algorithm runs continuously as the data received is continuous. In this type of data, there is a predefined execution time, if the algorithm exceeds this time limit, a lag will be generated between collected data points and the start time of the Iot device being used to run the algorithm.

Categories based on presumptions:

Parametric: Dataset is assumed to follow some predefined distribution like Gaussian distribution or Beta distribution etc. eg. Hotelling’s t2 test assumes the data to follow a Gaussian distribution.

Non- parametric: No assumptions are made for the data. It is usually observed that for online change detection, non-parametric methodsare effective and produce better results on the benchmark datasets.

Multivariate|approaches often assume|that each|example is drawn|from a multivariate process. Thus, we need not assume that the features are|independent. Multivariate change detection attempts to model a multivariate process by means of a function to evaluate the fit of new data (an example or a batch) to that model.

The change detection algorithm used is generated from the perspective of likelihood as a general|framework and is developed comparing with the two popular change detection algorithms, K-L Divergence and Hotelling’s t2-test. Each of these algorithms have their own limitations and the algorithm used, SPLL surpasses these algorithms in both normalised and non-normalised data, for less clusters as well as for more number of clusters.

For making this algorithm compatible with the online change detection, a sliding window method is used. Two windows, baseline window and current window are selected and compared using the Semi parametric log likelihood algorithm. This report also compares two cases, fixed and moving windows and proposes an adaptive baseline window based on the following metrics.

Metrics used:

In-control run length(IC-RL): In-control run length is the run length of number of data points which are assigned to be normal (no change observed by the algorithm) after a change is induced. This can also be explained as the number of deviated data points, the algorithm requires to check before raising an alert. Statistically, this can be explained as suppose Lt is the t-th test statistic of an in-control data stream {Z i } and a change is declared when Lt exceeds the threshold h (where a change has been induced). The in-control run Length(IC-RL) is defined as

IC − RL = min{t : L t > h} − 1

AUC-ROC : Receiver Operating Characteristic (ROC) curve is a performance measure for multiclass classification problems. The y - axis defines the True Positive Rate and the x - axis defines the False Positive Rate. The Area Under Curve (AUC) is the area under the ROC curve. If the Area under the curve is more than 0.5, the model is more likely to distinguish positive class values from the negative class values.

**PROBLEM STATEMENT**

Fruits like mangoes require constant monitoring and are vulnerable to environmental changes. It is difficult and expensive to monitor all the mangoes, in inventory or in trucks while transporting. This demanded an effective solution reducing the effort as well as labour required for maintaining the fruits in these facilities.

If any change is detected, the alarms are needed to be synced with the users (here, supervisors controlling the quality of fruits) using an blockchain implementation. In case of changes in storage facilities, any detected change should be notified to the supervisor and in case of transportation, any change detected should be notified to drivers or helpers in the truck.

The solution to this problem should be a long term solution which should not be upgraded every once a while and should be cost effective. Since the fruits are stored in closed storage, the solution should involve wireless implementation of resources and since frequent manual checking can further change the environment of the stored fruits, the solution should consider low maintainability, i.e the solution should be generalized for all possible cases and should be useful even in unpredictable conditions.

**OBJECTIVES**

* The change detection algorithm is to be used in an IoT device for which our algorithm needs to be fast as well as efficient.
* The algorithm used should be a non-parametric algorithm so that it is independent of any presumptions of the data following gaussian or any other distribution.
* The algorithm used should be applicable for online multivariate data. The data expected to be received from the sensor nodes are live multivariate data. So our algorithm should not consider the availability of the complete dataset beforehand for change detection.
* The time taken for the change detection algorithm to run and be ready to take the input of the next data point should be less than the time gap between the consecutive sensory readings.
* The algorithm is to be implemented in an IoT device which might use algorithms like Arduino or Raspberry Pi, so our algorithm needs to be computationally less expensive.
* The algorithm should be accurate. Since our change detection alert is going to be sent through a blockchain. So reducing the false alerts is a necessary factor for our algorithm.

**LITERATURE REVIEW**

Sensor nodes in Wireless Sensor Networks (WSNs) are used for collecting data from different locations, and are capable of connecting with other nodes forming a wireless network. Each cluster has a cluster-head node which possesses small computational power for computing small operations on the data collected from the sensor nodes.

Nodes are installed in different locations for a group analysis of the batches of mangoes. There are different sensor nodes installed like nodes for recording temperature, humidity, ammonium percentage, sulfur percentage etc for an collective analysis of different factors corresponding to each batch of fruits in the container.

Data for base window:

A batch of mangoes is used to collect the data for use as the base data (the data with which the sensor data are to be checked). This is done by installing multiple sensory nodes in a container. The keeping the fruit batches inside the container and leaving the whole set un tampered. This generates frequent data which can be used as a base data for checking the changes in the received data from real time sensor nodes of storage systems of big companies of truck facilities.

Multivariate log likelihood methods:

The methods in this category monitor the distributions of two windows of data. The basic construction involves a reference window composed of old data, and a detection window composed of new data. This can be achieved with a static reference window and a sliding detection window, or a sliding pair of windows over consecutive ob-

servations. The old and new windows can be compared with statistical tests, with the null hypothesis being that both windows are drawn from the same distribution.

Non-parametric and parametric methods:

The methods of likelihood estimators are of two types based on presumptions about the data distributions. Non-parametric methods like KL divergence and SPLL methods do not consider the data to follow any specific distribution for which these methods work good for all sorts of data distributions. Parametric methods like Hotelling’s t2-test assume that the data follows gaussian distribution. This method specifically checks based on t2 test on the squared Mahalanobis distance on the live multivariate data. For this problem statement, non-parametric methods are used for change detection.

**METHODOLOGY**

1. Clustering the data:

The multivariate data is clustered to get the mean and standard deviation of the sample data. The distribution of this data can be obtained through the Expectation Maximization (EM) family of algorithms. But since the algorithm is to be implemented in an IoT device, the algorithm should be computationally less expensive for which the K-Means algorithm is used to cluster the dataset. Now, the mean and variances of each cluster is calculated and the distribution is estimated. This provides an approximate estimate of the clustered data with less computational requirements.

1. Semi-parametric Log-Likelihood method:

This is a non-parametric log likelihood method comparing the deviation of each data point from the current window. Probability distribution of each cluster is calculated and their mean and covariance matrices are stored as μi and Σi respectively. In the likelihood estimation, the Mahalanobis distance of each data point in the current window is calculated with the probability distributions of clusters of the base window, the likelihood terms will not factorize because of the sum. This is why an upper bound is used and is replaced with the likelihood terms. The square of Mahalanobis distance of the data point is calculated for each data point from each cluster distribution and the maximum value is considered.

Then the mean of all the distance values of all the data points from the current window is considered. Using chi square statistics, if the current window distribution came from the base window distribution, then the squared Mahalanobis distances will satisfy the chi-squared test for n degree of freedom, n being the number of features in the multivariate data.

1. Fixed and sliding base window:

The base window is the window with which the data points are being compared for any changes in the distribution. There are two options as to whether our base window can be fixed or can be sliding based on no change data points. In fixed sliding base window, the base window is fixed and the data points are compared with the base window data. So after a certain time interval, the cluster-head node needs to reset and consider a new base window value.

This can be an issue specific to this problem statement. As in a day, there are certain changes based on the trend of the data, for example the temperature during the day is not equal to the temperature during the night. If a fixed sliding window is used this might create false alerts. So a sliding base window method is used which will adaptively slide the base window if for a certain data point, there is no change detected. Let's assume there is no change detected for m points and consider n as a threshold (greater than or equal to the base window size) for sliding the base window to these no change data points.

If m > n, slide the previous base window to the new m data points.

else, use the previous base window for change detection.

**ALGORITHM IMPLEMENTATION**

The algorithm is implemented in the following steps:

* Base window and current window are declared in the dataset based on parameter, window size.
* All the visual as well as computational parameters are declared and initialized.
* The Base window dataset is first clustered into ‘K’ clusters using gaussian mixture models. But since our objective is to provide a fast change detection algorithm, we are using the K- means algorithm for clustering.
* Mean and variance of each cluster is calculated. Then the covariance matrix is calculated taking weights of each cluster proportional to the number of elements in the corresponding cluster.
* Now, the partial inverse of the covariance matrix is calculated using the ‘pinv’ function from numpy library.
* This matrix is used to calculate the squared Mahalanobis distance of each current window datapoint from the nearest cluster center.
* The mean of the squared Mahalanobis distance of all data points is taken and a statistical chi-square test is used to compare the p-value for n degree of freedom.
* The change variable is updated based on the result from the above algorithm. The sliding of the base window is decided based on the threshold values and the change values.
* The above steps are repeated for the next data point.

**WORK DONE SO FAR**

* A semi parametric log likelihood method is selected among multiple methods to use as the change detection algorithm for the problem statement.
* A code for the SPLL algorithm is written in python and applied on various offline datasets. The algorithm is then compared with KL divergence and Hotelling’s t2 test methods based on the ROC curve - AUC. Then the algorithm is tested with various K (number of clusters), M (window size), normalised and non-normalised data. The ROC curves are plotted for each permutation.
* A live demonstration is done in the following steps:
* The ‘animations’ module from matplotlib is used to prepare a live animation model of the proposed method.
* Random normal points are generated for a fixed number of times to create a base window. The display, window size, mean and standard deviation, change values, alpha, K etc are initialized.
* Plots are initialised. One for combined feature sets, rest subplots for individual features.
* Sliding or fixed base window conditions are checked and assigned based on the parameters initialised.
* Current window is slid accompanying the new data point randomly generated.
* The SPLL algorithm is run based on the base window, current window, K, window size and alpha parameters. The output is appended in an array of size 100.
* Then a function ‘data\_generator’ is run for generating a new random point. Periodically a change is induced in the generated data by changing the mean and standard deviation of the distribution from which the random points are drawn.
* Now, the animation tab is run to demonstrate a live implementation of the proposed model.

**WORK TO BE DONE**

* Noise reduction: We can use different filtering techniques to remove simple noise values which may be considered as outliers by our algorithm. Some filtering techniques may be Gaussian filtering or moving average or mean filtering etc.
* We can have an ensemble method for change detection of some univariate methods for some subspaces and use aggregation techniques like voting methods to improvise our algorithm. We can also use an ensemble of some multivariate change detection algorithms like KL divergence, Hotelling’s t2-test and SPLL method.
* A protective mechanism against false alarms could be proposed for implementing in a separate cluster head node, in which the change data from the executed cluster head node change detection algorithm is passed. This will be an offline mechanism in which the algorithm will check for any possibility of false alarm and if the output is likewise, it will reset the alarm.
* An adaptive threshold mechanism could be implemented. The initial parameters defined are always fixed but a new feature can be added training on a train dataset. It will run the change detection algorithm based on some predefined parameters and then it will check for frequency of the alarms and using a quartile estimate, it can update the parameters. This method will possibly take some time to run in the train dataset after which it can automatically update the threshold values based on the quartile value input. This can be an interesting feature in which the user does not need to manually feed the parameters but only feed the quartile estimate and the algorithm can adaptively update all the required parameters based on the previous n data points.

***References***

[1] Yang Xiao-fei, Wu Xiao-bei, and Huang Jin-an, “*TAGPP: a tiny aggregation algorithm with preprocessing in local cluster”*, 2009 International Conference on Networks Security.

[2] Ludmila I. Kuncheva, *“Change Detection in Streaming Multivariate Data Using Likelihood Detectors”,* 2013 IEEE.

[3] Eric L. Bullock , Curtis E. Woodcock, Christopher E. Holden, *“Improved change monitoring using an ensemble of time series algorithms”,* 2019 Elsevier Inc*.*

[4] Lingzhe Guo & Reza Modarres, *“Two multivariate online change detection models”,* 2020 Journal of Applied Statistics

[5] William J. Faithfull, Juan J. Rodríguez, Ludmila I. Kuncheva, “*Combining univariate approaches for ensemble change detection in multivariate data”,* 2018 Elsevier B.V.

[6] Imran Khan, Joshua Z. Huang, Zongwei Luo, M.A. Masud, “*CPLP: An algorithm for tracking the changes of power consumption patterns in load profile data over time”,* 2017 Elsevier Inc.

[7] Lingyu Zhang, Jiabao Zhao, Wei Li, “*Online and Unsupervised Anomaly Detection for Streaming Data Using an Array of Sliding Windows and PDDs”,* 2019 IEEE

[8] Masoomeh Zameni, Zahra Ghafoori, Amin Sadri, Christopher Leckie, Kotagiri Ramamohanarao, “*Change Point Detection for Streaming High-Dimensional Time Series”*, DASFAA 2019

[9] X. Song, M. Wu, C. Jermaine, and S. Ranka, “*Statistical Change Detection for Multi-Dimensional Data*”, KDD ’07: Proc. 13th ACM SIGKDD Int’l Conf. Knowledge Discovery and Data Mining, pp. 667-676, 2007.

[10] Dang-Hoan Tran, “*Automated Change Detection and Reactive Clustering in Multivariate Streaming Data*”, arXiv:1311.0505v1 [cs.DB] 3 Nov 2013

[11] S.-S. Ho, “*A Martingale Framework for Concept Change Detection in Time-Varying Data Streams”,* Proc. 22nd Int’l Conf. Machine Learning (ICML), pp. 321-327, 2005.