Reading Elective Report

Paper - 1

"Learning with drift detection," in Proc. 17th Brazilian Symp. Artificial Intelligence, ser. Lecture Notes in Computer Science. Springer, 2004, Book Section, pp. 286–295

In this work we study the problem of learning when the class-probability distribution that generate the examples changes over time. We present a method for detection of changes in the probability distribution of examples. A central idea is the concept of ***context*: a set of contiguous examples where the distribution is stationary.** The idea behind the drift detection method is to control the online error-rate of the algorithm.

As a general rule, if a concept drift is detected the window size decreases, otherwise the window size increases. To detect concept changes the accuracy and the coverage of the current learner are monitored over time and the window size is adapted accordingly. In this work we assume that examples arrive one at a time. We consider the online learning framework.

In this framework when an example becomes available, the decision model must take a decision (e.g. an action). a sequence of examples, in the form of pairs $(\hat{x}_i, \hat{y}_i)$. For each example, the actual decision model predicts $\hat{y}_i$, that can be or True or False. For a set of examples the error is a **random variable from Bernoulli trials**. The Binomial distribution gives the general form of the probability for the random variable that represents the number of errors in a sample of n examples. For each point i in the sequence, the error-rate is the probability of observe False, p_i , with standard deviation given by $s_i = sqrt(p_i(1 - p_i)/i)$.

For sufficient large values of the example size, the Binomial distribution is closely approximated by a **Normal distribution** with the same mean and variance. Considering that the probability distribution is unchanged when the context is static, then the $1 - \alpha/2$ confidence interval for p with n $>$ 30 examples is approximately $p_i \pm \alpha * *s_i$. The parameter α depends on the confidence level. The drift detection method manages two registers during the training of the learning algorithm, p_{min} and s_{min}. Every time a new example i is processed those values are updated when $p_i + s_i$ is lower than $p_{min} + s_{min}$.

In the experiments described below the confidence level for warning has been set to 95%, that is, the warning level is reached if $p_i + s_i \geq p_{min} + 2 * s_{min}$. The confidence level for drift has been set to 99%, that is, the drift level is reached if $p_i + s_i \geq p_{min} + 3 * s_{min}$.

Suppose a sequence of examples where the error of the actual model increases reaching the warning level at example k_w, and the drift level at example k_d. This is an indication of a change in the distribution of the examples. **A new context is declared** starting in example k_w, and a new decision model is induced using only the examples starting in k_w till k_d.

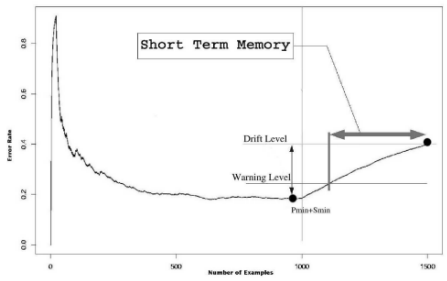


Fig. 1. Dynamically constructed Time Window. The vertical line marks the change of concept.

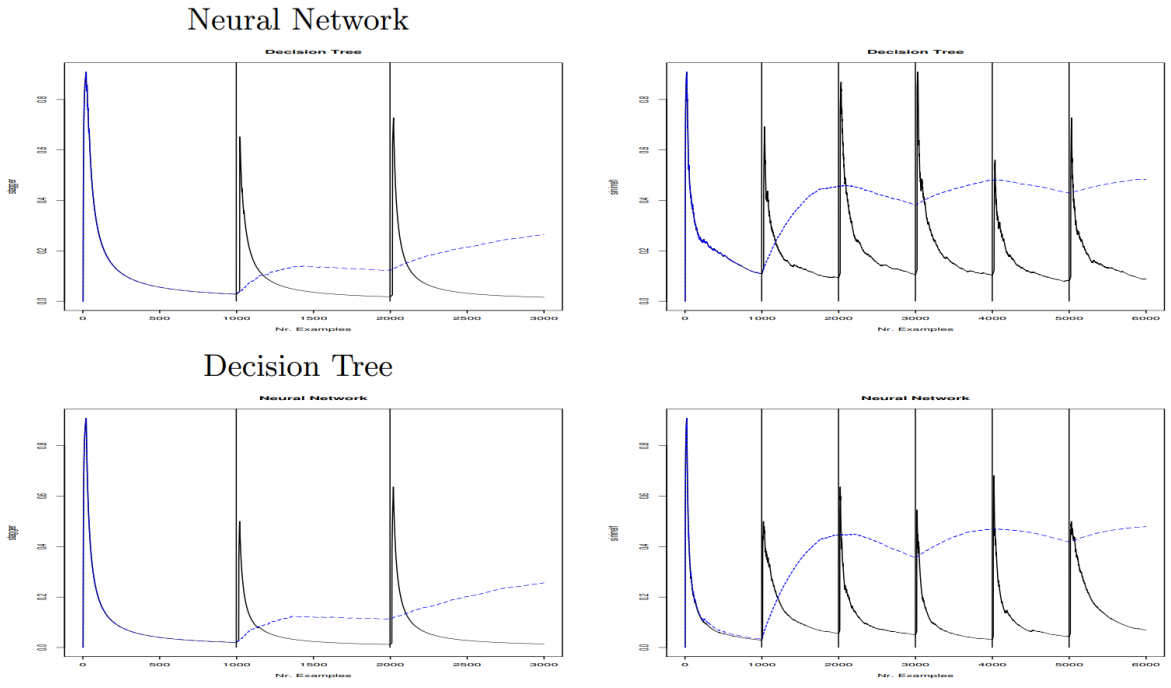
It is possible to observe an increase of the error reaching the warning level, followed by a decrease. We assume that such situations corresponds to a **false alarm**, without changing the context.

This method could be applied with any learning algorithm. It could be directly implemented inside online and incremental algorithms, and could be implemented as a wrapper to batch learners. The goal of the proposed method is to **detect sequences of examples with a stationary distribution**. We denote those sequences of examples as **context**. From the pratical point of view, what the method does is to choose the training set more appropriate to the actual class-distribution of the examples.

The following method has been implemented onto **8 Artificial Data Sets.** All the problems have two classes. Each class is represented by 50% of the examples in each context. Each dataset embodies at least two different versions of a target concept. Each context defines the strategy to classify the examples. Each dataset is composed of 1000 random generated examples in each context.

1. **SINE1**. Abrupt concept drift, noise-free examples.
2. **SINE2**. The same two relevant attributes.
3. **SINIRREL1**. Presence of irrelevant attributes.
4. **SINIRREL2**. Same classification function of SINE2 but the examples have two more random attributes
5. **CIRCLES**. Gradual concept drift, noise-free examples.
6. **GAUSS**. Abrupt concept drift, noisy examples.
7. **STAGGER**. Abrupt concept drift,symbolic noise-free examples.
8. **MIXED**. Abrupt concept drift, boolean noise-free examples.

In this paper, we have shown the effect of the proposed drift detection method on the generalization capacity of each learning algorithm. the results of the application of the drift detection method with the results without detection are shown below.



Three learning algorithms and Two artificial datasets.

Nevertheless, the differences are more significant with the neural network and the decision tree.

This method was then applied on the **Electricity Market Dataset**, collected from the Australian New South Wales Electricity Market. In this market, the prices are **not fixed** and are affected by demand and supply of the market. The prices in this market are **set every five minutes**. The interest of this dataset is that it is a real-world dataset. We dont know when drift occurs or if there is drift.

The **ELEC2 dataset** contains **45312 instances** dated from 7 May 1996 to 5 December 1998. Each example of the dataset refers to a period of **30 minutes**, i. e. there are **48 instances for each time period of one day**. Each example on the dataset has **5 fields**,

- The day of week
- The time stamp
- The NSW electricity demand
- The Vic electricity demand
- The scheduled electricity transfer between states and the class label.

The class label identifies the change of the price related to a moving average of the last 24 hours. The class level only reflect deviations of the price on a one day average and removes the impact of longer term price trends.

Dataset	*Perceptron*		Neural Network		Decision Tree	
	No Detection	Detection	No Detection	Detection	No Detection	Detection
STAGGER	0.048	**0.029**	0.351	**0.002**	0.265	**0.016**
SINE1	0.126	**0.115**	0.489	**0.019**	0.490	**0.081**
SINIRREL1	0.159	**0.139**	0.479	**0.068**	0.483	**0.088**
SINE2	0.271	**0.262**	0.492	**0.118**	0.477	**0.100**
SINIRREL2	0.281	0.281	0.477	**0.059**	0.485	**0.084**
MIXED	0.100	0.111	0.240	**0.065**	0.491	**0.465**
GAUSS	0.384	0.386	0.395	**0.150**	0.380	**0.144**
CIRCLES	0.410	0.413	0.233	**0.225**	0.205	**0.109**

Table 1. Final learning algorithm error rate with and without drift detection active.

We have two problems to deal with this dataset.

- Predict the changes in the prices relative to the last day.
- Predict the changes in the prices relative to the last week of examples recorded.

the learning algorithm, the implementation of CART available in R, learns a model from the training data. We have used the proposed method as a wrapper over the learning algorithm. After seeing all the training data, the final model classifies the test data.

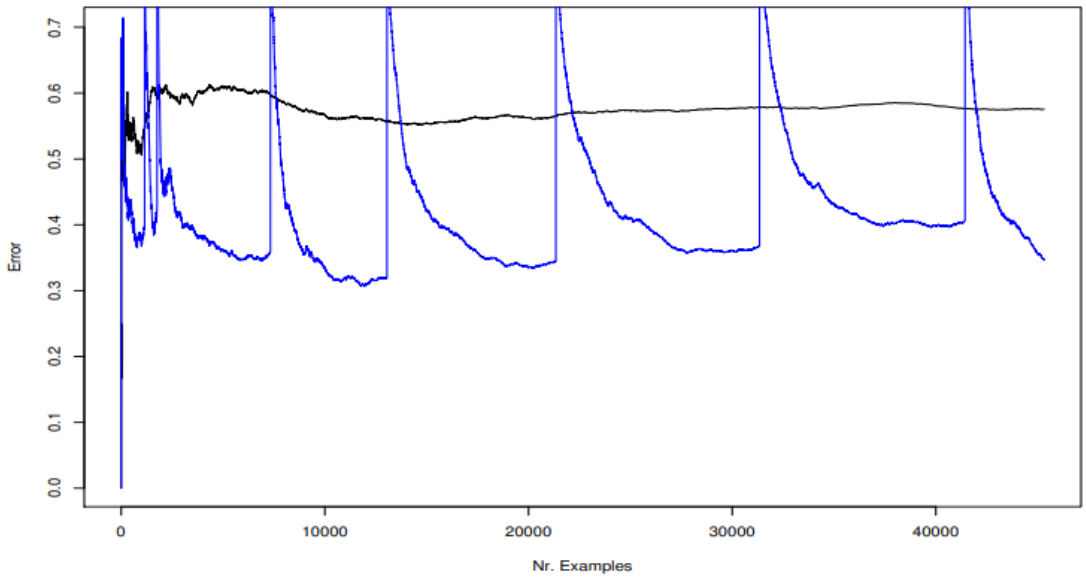


Fig. 3. Trace of the on-line error using the Drift Detection Method applied with a Decision Tree on ELEC2 dataset.

Test Set	*Lower Bound*	*Upper Bound*		Drift Detection
		All Data	*Last Year*	
Last Day	0.104	0.187	0.125	0.104
Last Week	0.190	0.235	0.247	0.199

Table 2. Results of the error rate for two different Test Sets.

To conclude, I learnt a method for detection of concept drift in the distribution of the examples. The method is simple, with direct application and is computationally efficient. The Drift Detection Method can be applied to problems where the information is available sequentially over time. The method is independent of the learning algorithm. It is more efficient when used with learning algorithms with greater capacity to represent generalizations of the examples. This method improves the learning capability of the algorithm when modeling non-stationary problems. We intend to proceed with this research line with other learning algorithms and real world problems. We already started working to include the drift detection method in an incremental decision tree. Preliminary results are very promising. The algorithm could be applied with any loss-function given appropriate values for α. Preliminary results in regression domain using mean-squared error loss function confirm the results presented here.

Paper - 2

Concept Drift Detection in Data Stream Clustering and its Application on Weather Data

The paper mentions about a proposed framework which is capable of explicit concept drift detection and cluster evolution analysis. Relationship between concept drift and the occurrence of physical events has been studied by applying the framework on the weather data stream. Experiments led to **The conclusion that the concept drift accompanied by a change in the number of clusters indicates a significant weather event.**

Capability to process the data in a single scan, requirement of online and incremental updation of the models, adaptation to the concept changes etc. are some challenges while designing learning algorithms for data streams. Clustering is an important machine learning task applied on data streams to gain useful insights into the natural groupings in data.

The main advantage of data stream mining is that it assumes the data source or the process generating the stream is not stationary. According to the changes in the environment that produces the stream, the underlying data distributions change over time. Consequently, these changes might affect the inter relationship between input and output variables leading to **'Concept Drift'**. In these situations, the learned model becomes obsolete and the prediction accuracy reduces considerably. **Weather data** is a classic example where the concept can change over time.

This paper proposes a framework for the **online clustering of data streams**. It performs **concept drift detection** and **cluster evolution monitoring** to generate a warning on the dynamic changestaking place in the environment of the stream. Studies revealed that concept drift accompanied by clustering structure changes often imply important physical events. The framework is used to study the interrelationship between the changes in the clustering structure and the evolution of the **south-west monsoon**.

The proposed framework has components to-

- Cluster the stream online
- Detect concept changes
- Track the evolution of clusters.

Before performing the clustering, the best value for the **'number of clusters', k** is computed dynamically. This is done with the intention to **identify the changes** in the clustering structure for the recently arrived data. As the source of the data is highly dynamic, the clustering structure also might exhibit a corresponding change and fixing the value of k limits the ability to capture such changes in the clustering structure.

THE PROPOSED FRAMEWORK

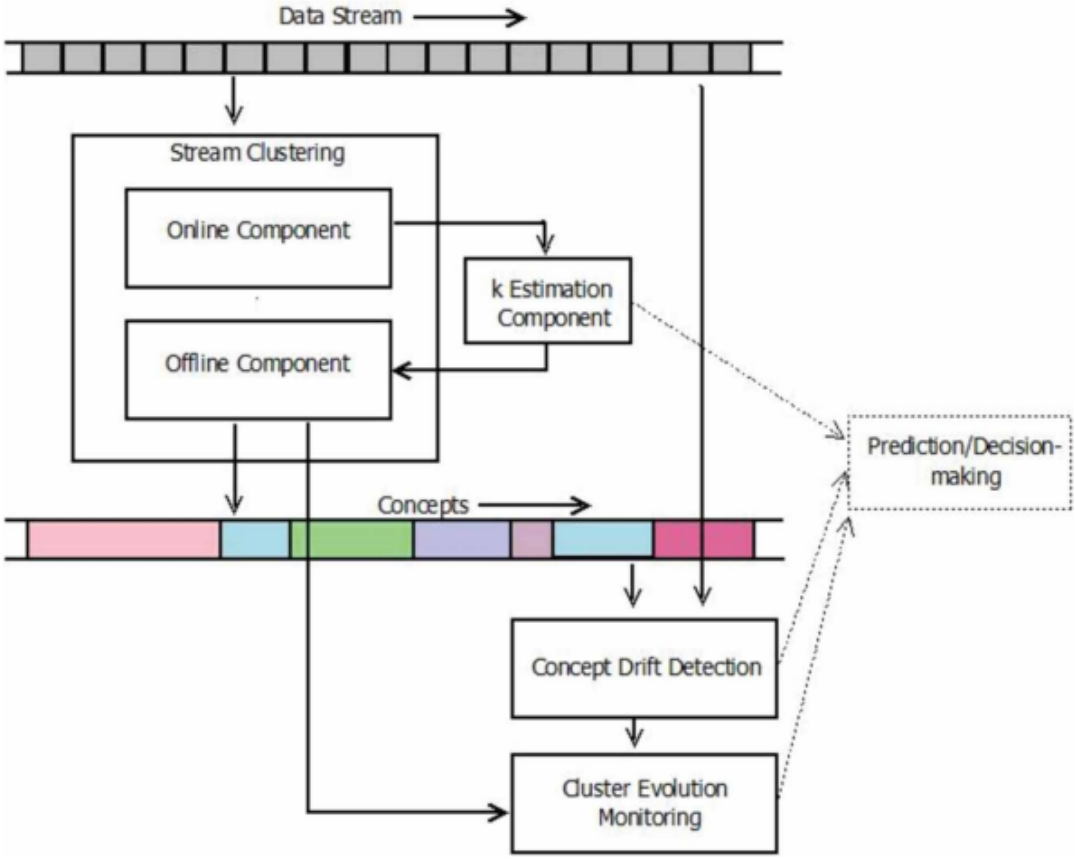
Online stream clustering is performed using a two-level clustering algorithm.

- Online streaming component- Generates the synopsis of the stream
- Offline streaming component- Generates the clustersfrom this synopsis.

Computation of the best value of k is performed between these two phases to generate that many clusters during the offline clustering. a concept is defined as the probability distribution P(X) where X is the random variable over vectors of attribute values. Clustering or the partition produced at a particular time point is nothing but a representation of the data distribution P(X) at that moment. Clustering can be considered as a probability mixture model with each cluster representing one component of the mixture. Since CluStream uses k-means for offline clustering, the clusters generated are convex in shape and follows a normal distribution. So, the nth component of the mixture model can be written as:

$$\varphi_n\left(x \mid \mu_n, \Sigma_n\right) \equiv \frac{\exp\left\{-\frac{1}{2}(x-\mu_n)^T \Sigma_n^{-1}\left(x-\mu_n\right)\right\}}{\sqrt{\det\left(2\pi\Sigma_n\right)}}$$

Figure 1. Overview of the proposed framework



Since the clustering depicts the data distribution, the clustering generated by the offline component of the framework is treated as the concept learnt from the data. At a time only one concept is active and the fitness of this concept to the recently arrived data is always being monitored. Concept drift detection module is used for this purpose. Current concept and the samples retrieved from the stream are the input to this module. Whenever the change is detected, a new clustering is generated from the recent data items and the current concept is replaced by this new concept. It has to be noted that clusters undergo various transitions during these concept drifts. Cluster evolution analysis is done to understand the nature of these transitions. Information obtained from the concept drift detection module, k estimation module and the cluster evolution analysis can be combined together to support the prediction process.

1. Stream Clustering-

A brief discussion on the stream clustering algorithm used in this framework is-

Clustering is a method commonly used for statistical analysis of data. It is defined as the task of grouping a set of data points into clusters such that points within one cluster are similar to each other and points from different clusters are dissimilar. In this paper, stream clustering is done using **CluStream**. CluStream has good clustering accuracy and an in-built capability to deal with evolving data streams. It divides the clustering process into online and offline components.

- **Online component-** Generating micro-clusters
  Micro-clustering is the method of storing the summary statistics of the stream. A micro-cluster is stored as a five element tuple - altogether representing the summary of data points in that cluster. This summary serves the purpose of calculating the characterising features of the cluster.

  Radius, diameter and summary of the timestamp information of the members is also stored in each micro-cluster. Removal of some micro-clusters from the memory is necessary as part of the evolution of the

stream, where we use this information. On the arrival of each data record of the stream, it's similarity to the existing micro-clusters is computed. If found similar, it is added to the most similar micro-cluster, otherwise a new micro-cluster is formed to represent this data record, giving an impression that stream is evolving.

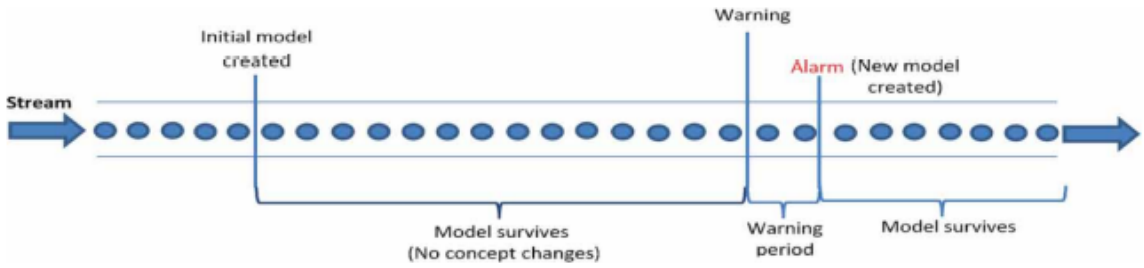
- **Offline component-** Generating macro-clusters from micro-clusters.
  Macro-clustering phase is the one which creates the real outcome of the stream clustering process. , It retrieves the micro-clusters relevant to that time period and does an offline clustering. During this clustering phase, micro-clusters are treated as single data points represented by their centres.

  Initial cluster centres or the seeds are not selected randomly, but the more weighted k micro-clusters are chosen asseeds. Further, we have introduced a modification to this algorithm. The value of 'k' in macro-clustering phase is not fixed, it is calculated dynamically based on an algorithm.

In CluStream algorithm, macro-clustering process is initiated upon user request and it generates macro-clustersrelevant for the period requested by the user. Upon creation of a group of macro-clusters, that particular clustering is treated as the active model as long as the underlying data distribution persists, or in other words, until the same concept survives.

Page-Hinkley Test (PHT) is adopted here, which monitors the stream continuously to find out the probable concept changes. First a warning is given and if still the stream continues to deviate from the current concept, an alarm is triggered to declare the concept drift. When a concept drift is flagged by this algorithm, next macro-clustering process is initiated. Before running the macro-clustering phase, value of 'k' is estimated dynamically. On creation of the new k macro-clusters, the new model is establishe

Figure 2. Overview of the concept drift detection process



- Online maintenance of micro-clusters;
- Storing the micro-clusters in memory;
- Page-Hinkley Test to detect concept drift.

And when concept drift detection algorithm flags an alarm, the following four processes are executed:

- Extraction of the micro-clusters relevant to the warning period i.e., the time period between warning flag and alarm flag;
- Computation of the value of k, i.e., the number of macro-clusters;
- Creation of the new model;
- Recalculation of the Page-Hinkley Test (PHT) parameters.

2. Calculating Number of Clusters Online-

Data stream clustering algorithms based on the principle of k-means usually work on the assumption that the number of clusters 'k' is provided by the user and it is fixed throughout the stream. The CluStream algorithm also works on this assumption. Change in the number of clusters is an indication of a change in data distribution, which in turn implies a possible drift in the prevailing physical conditions.

To find the best value of k, there should be a method to assess the relative quality of different data partitions. Comparison studies also reveal that **Simplified Silhouette** is the best method for assessing the quality of a data clustering. Silhoutte value is calculated as following- Consider x_i is an element of cluster C_a, and $a(x_i)$ is the average dissimilarity of x_i to all other elements of C_a. For each cluster C_b other than C_a, compute the average distance between x_i and the elements in C_b. The cluster having the lowest value for this average distance is called the neighbouring cluster of x_i and let this lowest value be denoted as $b(x_i)$. It is generally observed that a good clustering will have low value for $a(x_i)$ and high value for $b(x_i)$. Silhouette can be measured as-

$$s\left(x_i\right) = \frac{b\left(x_i\right) - a\left(x_i\right)}{\max\left\{a\left(x_i\right), b\left(x_i\right)\right\}}$$

the value of $s(x_i)$ will be within the interval [0,1]. Bigger value for $s(x_i)$ implies better clustering. $a(x_i)$ is the distance between x_i and its own cluster's centroid. Similarly, to calculate $b(x_i)$, distances between xi and all other cluster centroids are taken. This variant of a silhouette is called Simplified Silhouette (SS).

Average of all the $s(x_i)$ over i = 1, . . . , N is taken as the SS value

$$SS = \frac{1}{N}\sum_{i=1}^{N} s\left(x_i\right)$$

Partition with the highest value of SS is taken as the best clustering.

3. Concept Drift Detection-

CluStream has been built with the capabilitiesto handle an evolving data stream. But it does not make use of any explicit change detection procedure. Explicit change detection is desirable because it helps to deal with the change in a timely manner and recover quickly from the performance drop. Also, this framework needs to assess the clustering structure changes on the identification of a concept drift, which is possible only if drift issignalled explicitly. Page-Hinkley Test (PHT) is a standard procedure for change detection. A basic principle behind change detection methods is to track how well the existing model fits the recently arriving data points. If the recently arriving data points deviate considerably from the current model, it indicates that the model is becoming unfit to represent the current data partition. In other words, the concept is changing and model needs a reasonable update. Page-Hinkley Test continuously monitors a parameter which can represent the change. In clustering, the average distance between the data records and their closest cluster centres is the parameter being monitored. Page-Hinkley Test is the method to be used for explicit change detection in streaming environments. Let D_i be the distance variable whose value is being monitored by Page-Hinkley test. D_i is the distance of data record i to its closest cluster centre in the current data partition. A cumulative variable m_T is calculated as:

$$m_T = \sum_{i=1}^{T}\left(D_i - \overline{D_i} - \delta\right)$$

where:

$$\overline{D_i} = \frac{\sum_{j=1}^{i} D_j}{i}$$

- m_T is the sum of deviation between the observed variable D_i and its mean till moment T
- δ is the tolerance level.
- Two threshold values λ_A and λ_W are used to denote
  the alarm threshold and warning threshold respectively. Obviously, λ_A should be set greater than λ_W.

Page-Hinkley Test gives a warning when the difference between m_T and it's minimum M_T = Min(m_i, i = 1, . . . , T) becomes greater than the warning threshold λ_W. i.e., when m_T - $M_T > \lambda_W$. Similarly, it triggers an alarm when this value becomes greater than the alarm threshold λ_A.

The process of macro-clustering, which isresponsible for creating the new model, isinitiated as a concept drift is detected. the proposed framework just notes the warning period and extracts only those micro-clusters which are relevant to this time interval. Extracted micro-clusters are then fed as input to the macro-clustering phase. The whole set of PHT parameters, including the threshold values are recalculated after creating the new model.

4. Cluster Evolution Analysis-

Besides the changesin the number of clusters,substantial cluster transitions also occur during concept drifts. Cluster transitions include the disappearance of long-lived clusters, creation of new clusters, merging and splitting of existing clusters and so on. Cluster evolution analysisis performed as part of the experiments, to uncoversuch cluster level transformations occurring at concept change points. The literature proposes a framework named MONIC, to monitor the cluster transitions. It basically compares the clusters in two consecutive clusterings and finds out the differences and similarities between them, in terms of their internal and external transitions. As part of this study, experiments are conducted particularly to unveil the external transitions, namely - absorption, survival, split, disappearance and creation of a new cluster.

THE EXPERIMENT ON REAL LIFE DATA

Experiments are conducted to study the clustering behavior of weather data and its relationship to the physical phenomena occurring in the environment. Onset and withdrawal of south-west Monsoon and thunderstorms occurrences are the weather phenomena selected for the study.

In Cochin, Kerala, normally the onset of southwest monsoon happens during June first week, and it withdraws during the beginning of October. Also, unpredictable rain and thunderstorm are common during the summer season. Data collected in the years 2016 and 2017 are used for this study.

The following work is done on the data-

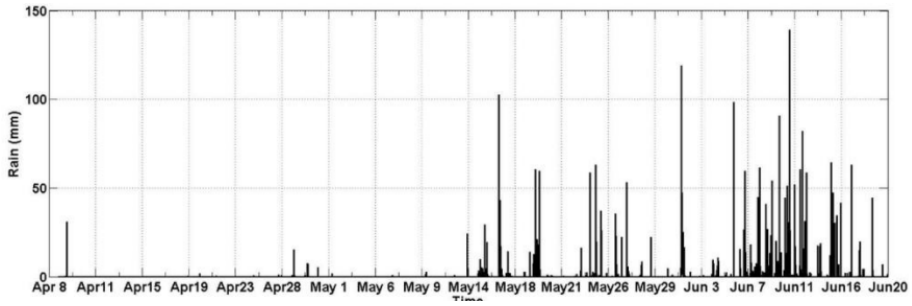
1. The Min-Max procedure is performed to normalize the data values to the range [0,1].
2. The first model will be created in an offline way using the initial 5000 samples of the stream. K-means clustering is the algorithm used in this stage.
3. Thereafter, the model is updated online, along with the stream.
4. Online maintenance of micro-clusters in CluStream algorithm makes sure that the total number of micro-clusters in main memory does not exceed a predetermined limit at any point of time.
5. In these experiments, this limit is fixed to 200.

Automatic Weather Station in ACARR collects most of the weather parameters at three different height levels: 2 meters, 20 meters, and 30 meters. Hence, there is a redundancy in information as far as prediction is concerned. All these parameters might not be relevant for prediction. A dimensionality reduction technique was applied in the beginning to find out the most relevant dimensions or parameters.

The parameters helpful for the follwoing are listed in the table below-

Rain related studies	***Thunderstorm related studies***
Temperature	Temperature
Cloud Radiation	Humidity
Solar Radiation	Wind speed
Net Radiation	
Wind Speed	
Wind Direction	

1. **Onset of South-West Monsoon**



As it is evident from this figure, there is a pre-monsoon shower starting from May 14th onwards. Temperature, Solar radiation and most of the weather parameters show a significant difference in distribution during this period. A change in all the weather parameters can be observed in the week of 16th to 20th May. During the same time, Pre-Monsoon shower started and it prolonged till the monsoon onset during June first week.

2. **Withdrawl of South-West Monsoon**

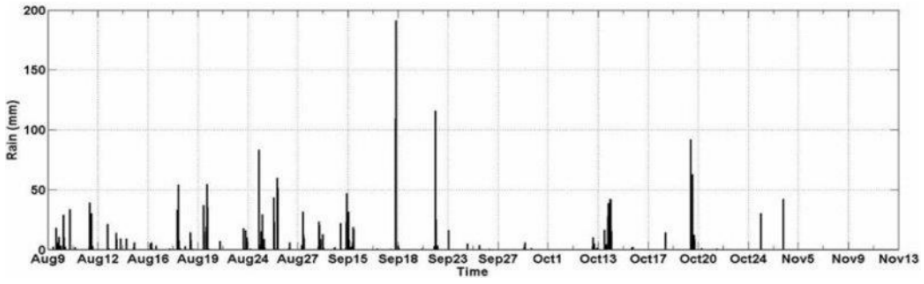
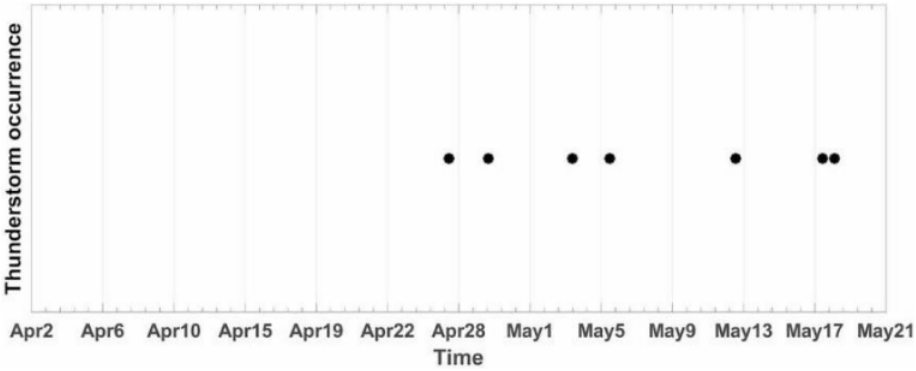


Figure shows the rainfall received during the southwest monsoon withdrawal period. Concept changes are found relatively frequent compared to the monsoon onset period. These results also support the fact that a significant change in clustering is an indication of a significant weather event.

3. **Thunderstorms**



In Kerala, during the summer season, it is common to get unpredicted rainfall with thunderstorm and lightning. Weather parameters show considerable changes during thunderstorms as well. Figure shows the occurrence of thunderstorms during the summer season. Black dots in the figure denote thunderstorm observations. Temperature, humidity and wind speed are identified to be the most prominent weather parameters that show variations during a thunderstorm. Datastream containing these three parameters is taken to check the clustering structure behavior and its relation to the thunderstorm occurrence.

International Institute of Information Technology
Reading Elective for Semester - 7,
Under the guidance of Prof. Uttam Kumar,
And Ms. Deeksha Agarwal.

END OF REPORT

By,
Fahed Shaikh,
IMT2019079.