ISSN 0798 1015

REVISTA ESPACIOS

Vol. 38 (Nº 59) Year 2017. Page 27

# ClusCTA-MEWMAChart: A new clustering-based technique to detect Concept Drift in the presence of noise

## ClusCTA-MEWMAChart: Una nueva técnica basada en clustering para detectar concept drift en presencia de ruido

Sonia JARAMILLO-VALBUENA 1; Jorge Mario LONDOÑO-PELÁEZ 2; Sergio Augusto CARDONA 3

Received: 15/08/2017 • Approved: 12/09/2017

### Content

**ABSTRACT:**
Many real-world applications generate data streams. Typically, data evolves over time and must be processed on-the-fly without the need for long term storage or reprocessing. In machine learning, the inherent variability or change over time of streams is referred to as concept drift. This phenomenon creates new challenges not present in classical machine learning techniques. In this paper, we present a new clustering-based technique to detect Concept Drift on data streams, named ClusCTA-MEWMAChart. We compare our algorithm experimentally with 4 different methods to detect concept drift from data streams and determine their robustness in the presence of noisy data. We conducted a set of experiments on synthetic datasets. The results show that the proposed approach has good performance.
**Key words** Concept drift; Classification; Data Stream Mining; adaptive learning

**RESUMEN:**
Muchas aplicaciones del mundo real generan grandes cantidades de datos que son continuos. A esto se le conoce como streams de datos. Típicamente, los streams de datos evolucionan con el tiempo. Este fenómeno, conocido como concept drift, crea nuevos desafíos que no están presentes en las técnicas clásicas de aprendizaje automático, dado que los datos deben ser procesados sobre la marcha sin la necesidad de almacenamiento a largo plazo o reprocesamiento. En este artículo, se presenta una nueva técnica basada en clusters para detectar Concept Drift en streams de datos, llamada ClusCTA-MEWMAChart. Este algoritmo es comparado experimentalmente con 4 métodos diferentes para detectar concept drift en streams de datos y se determina su solidez en presencia de ruido. Se realizan experimentos en datasets sintéticos. Los resultados muestran que el enfoque propuesto tiene un buen desempeño.
**Palabras clave** Concept drift; clasificación; minería sobre streams de datos, aprendizaje adaptativo

# 1. Introduction

In recent years many applications generating huge volumes of data in streamed manner have become commonplace.  Network monitoring, phone records, social networks, *ATMtransactions,* sensor data, and biological applications are examples of sources of data streams. Streaming data pose challenging research problems in order to respond to tasks such as statistics maintenance, storage, and real time querying and pattern discovery (Gama J.). Most traditional data mining techniques have to be adapted to analyze streaming data on the fly, normally trying to avoid storing the data in the stream (Le, Stahl, Gomes, Medhat Gaber & Di Fatta, 2014). The processing when the underlying data model changes over time is difficult because it must consider the presence of noise (Fang Chu, 2005) and the evolving nature of the stream, and therefore, should be able to detect and adapt quickly to concept drifting.

The term "concept drift" means that the statistical properties of the target concept change arbitrarily over time (Widmer & Kubat, 1996) (Wang, Yu, & Han, 2010) (Minku & Yao, 2012), causing the previously constructed model to be inconsistent and requiring an update or to be replaced with a new one to prevent accuracy deterioration (Wang, Yu, & Han, 2010) (Dongre & Malik, 2014).

Kelly et al. (Kelly, Hand & Adams., 1999) indicate that depending on the nature of the concept drift, it can occur in three different ways:

> The prior probabilities of classes P(c1),..., P(ck) might change over time.
>
> Class-conditional probability distributions might change P(X|ci), i = 1,…, k over time. This kind of concept drift is referred to as virtual drift, because it is possible that P(X|ci) change without  affecting the previous classes (e.g symmetric movement to opposite directions).
>
> Posterior probabilities P(ci|X), i = 1,…,k might change. This kind of change is often known as real drift.

The concept drift can also be classified *based on*its speed. In particular the speed can be: sudden or gradual. The sudden drift (also referred to as concept change) represents irreversible and abrupt changes of samples of respective class. The gradual drift corresponds to slow and gradual changes over time (Gama, Zliobaite, Bifet, Pechenizkiy, & Bouchachia, 2014) (Dongre & Malik, 2014).

An important challenge for Concept Drift Detection algorithms is to differentiate the occurrence of outliers or *noise* in the data from actual changes in the target concept. The outliers refer to random deviations or *statistical anomalies* (Chandola, Banerjee & Kumar, 2009). There are some approaches designed to handle concept drift but only very few studies in the literature have documented the performance of these techniques in the presence of noise.

In this paper we propose a new clustering technique to detect Concept Drift on data streams, it is referred as ClusCTA-MEWMAChart. This approach uses multiple sliding windows and centroid tracking for maintaining enough knowledge about centroid behavior, and multivariate EWMA to report the occurrence of concept drift.

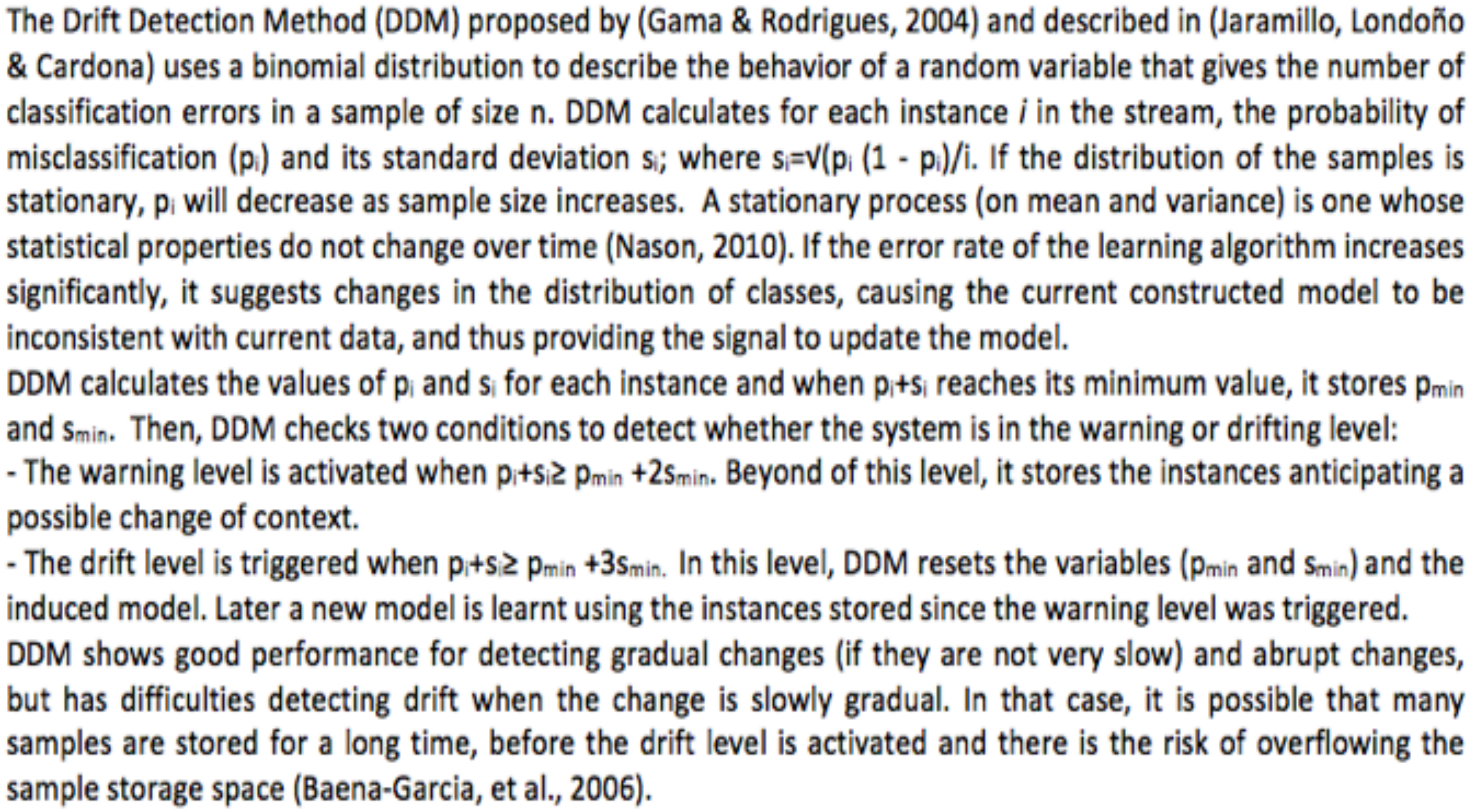
We compare our method with 4 different algorithms  for concept-drift detection: the Drift Detection Method (DDM) (Gama & Rodrigues, 2004),  the Early Drift Detection Method EDDM (Baena-Garcia, et al., 2006), a drift detection method based on Geometric Moving Average Test (Roberts, 2000) and the Exponentially Weighted Moving Average Chart Detection Method EWMAChartDM (Del Castillo, 2001) (Ross, Adams, Tasoulis & Hand, 2012) (Page, 1954) (Mouss & Sefouhi, 2004) .

The paper is organized as follows. Section 2 reviews related works on concept drift detection. Section 3 describes the technique we propose. Section 4 shows implementation details and model evaluation. The last section provides a summary of the findings in this work.

# 2. Related word

There are several approaches designed to detect concept drift. In this section we describe 4 known algorithms in the literature: DDM, EDDM, GeometricMovingAverageDM and EWMAChartDM.

## 2.1. Drift Detection Method (DDM)



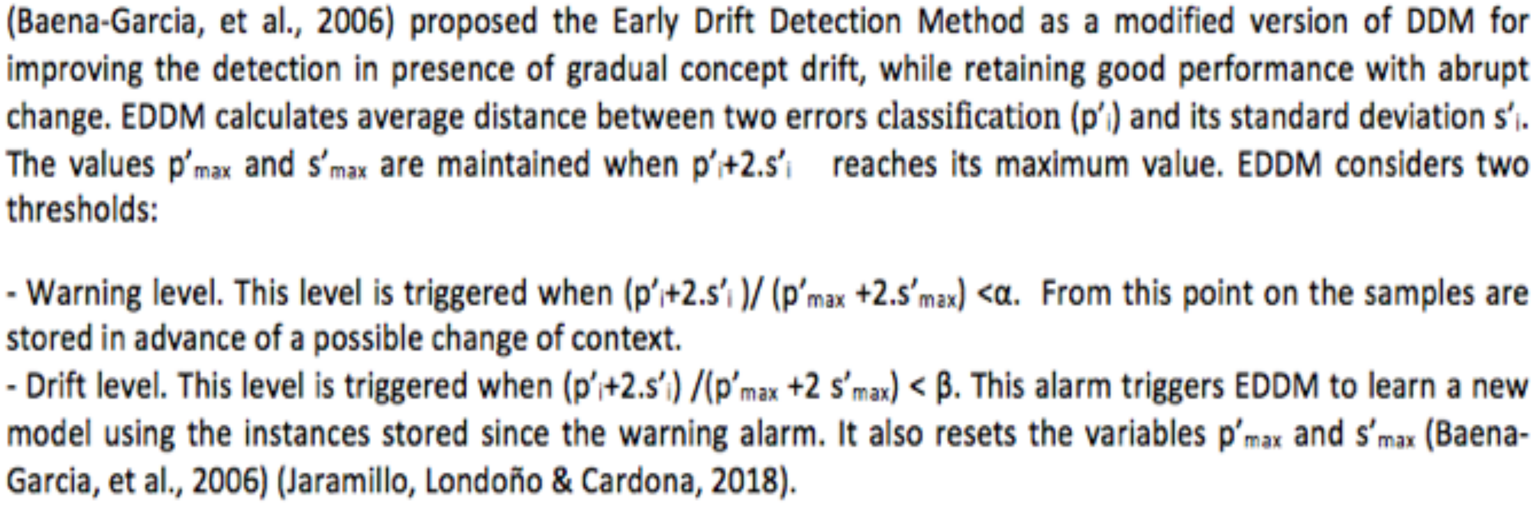
The Drift Detection Method (DDM) proposed by (Gama & Rodrigues, 2004) and described in (Jaramillo, Londoño & Cardona) uses a binomial distribution to describe the behavior of a random variable that gives the number of classification errors in a sample of size n. DDM calculates for each instance *i* in the stream, the probability of misclassification ($p_i$) and its standard deviation $s_i$; where $s_i=\surd(p_i\ (1 - p_i)/i$. If the distribution of the samples is stationary, $p_i$ will decrease as sample size increases. A stationary process (on mean and variance) is one whose statistical properties do not change over time (Nason, 2010). If the error rate of the learning algorithm increases significantly, it suggests changes in the distribution of classes, causing the current constructed model to be inconsistent with current data, and thus providing the signal to update the model.

DDM calculates the values of $p_i$ and $s_i$ for each instance and when $p_i+s_i$ reaches its minimum value, it stores $p_{min}$ and $s_{min}$. Then, DDM checks two conditions to detect whether the system is in the warning or drifting level:

- The warning level is activated when $p_i+s_i\geq p_{min} +2s_{min}$. Beyond of this level, it stores the instances anticipating a possible change of context.
- The drift level is triggered when $p_i+s_i\geq p_{min} +3s_{min.}$ In this level, DDM resets the variables ($p_{min}$ and $s_{min}$) and the induced model. Later a new model is learnt using the instances stored since the warning level was triggered.

DDM shows good performance for detecting gradual changes (if they are not very slow) and abrupt changes, but has difficulties detecting drift when the change is slowly gradual. In that case, it is possible that many samples are stored for a long time, before the drift level is activated and there is the risk of overflowing the sample storage space (Baena-Garcia, et al., 2006).

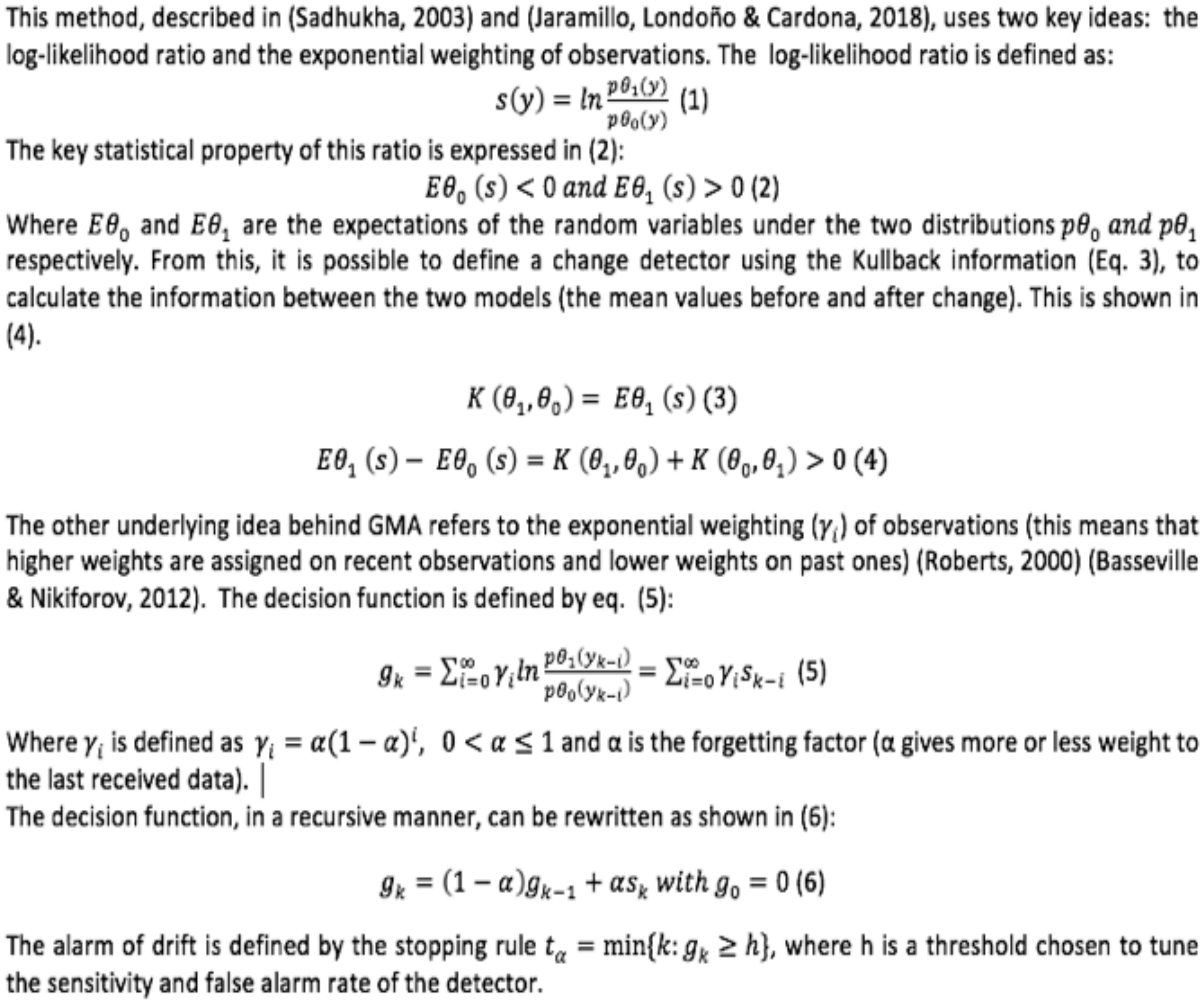
## 2.2. Early Drift Detection Method (EDDM)



(Baena-Garcia, et al., 2006) proposed the Early Drift Detection Method as a modified version of DDM for improving the detection in presence of gradual concept drift, while retaining good performance with abrupt change. EDDM calculates average distance between two errors classification ($p'_i$) and its standard deviation $s'_i$. The values $p'_{max}$ and $s'_{max}$ are maintained when $p'_i+2.s'_i$ reaches its maximum value. EDDM considers two thresholds:

- Warning level. This level is triggered when $(p'_i+2.s'_i\ )/\ (p'_{max} +2.s'_{max}) <\alpha$. From this point on the samples are stored in advance of a possible change of context.
- Drift level. This level is triggered when $(p'_i+2.s'_i)\ /(p'_{max} +2\ s'_{max}) < \beta$. This alarm triggers EDDM to learn a new model using the instances stored since the warning alarm. It also resets the variables $p'_{max}$ and $s'_{max}$ (Baena-Garcia, et al., 2006) (Jaramillo, Londoño & Cardona, 2018).

## 2.3. GeometricMovingAverage (GMA) Detection Method



This method, described in (Sadhukha, 2003) and (Jaramillo, Londoño & Cardona, 2018), uses two key ideas: the log-likelihood ratio and the exponential weighting of observations. The log-likelihood ratio is defined as:

$$s(y) = ln \frac{p\theta_1(y)}{p\theta_0(y)} \quad (1)$$

The key statistical property of this ratio is expressed in (2):

$$E\theta_0\ (s) < 0\ and\ E\theta_1\ (s) > 0\ (2)$$

Where $E\theta_0$ and $E\theta_1$ are the expectations of the random variables under the two distributions $p\theta_0\ and\ p\theta_1$ respectively. From this, it is possible to define a change detector using the Kullback information (Eq. 3), to calculate the information between the two models (the mean values before and after change). This is shown in (4).

$$K\ (\theta_1, \theta_0) = E\theta_1\ (s)\ (3)$$

$$E\theta_1\ (s) - E\theta_0\ (s) = K\ (\theta_1, \theta_0) + K\ (\theta_0, \theta_1) > 0\ (4)$$

The other underlying idea behind GMA refers to the exponential weighting ($\gamma_i$) of observations (this means that higher weights are assigned on recent observations and lower weights on past ones) (Roberts, 2000) (Basseville & Nikiforov, 2012). The decision function is defined by eq. (5):

$$g_k = \sum_{i=0}^{\infty} \gamma_i ln \frac{p\theta_1(y_{k-i})}{p\theta_0(y_{k-i})} = \sum_{i=0}^{\infty} \gamma_i s_{k-i} \quad (5)$$

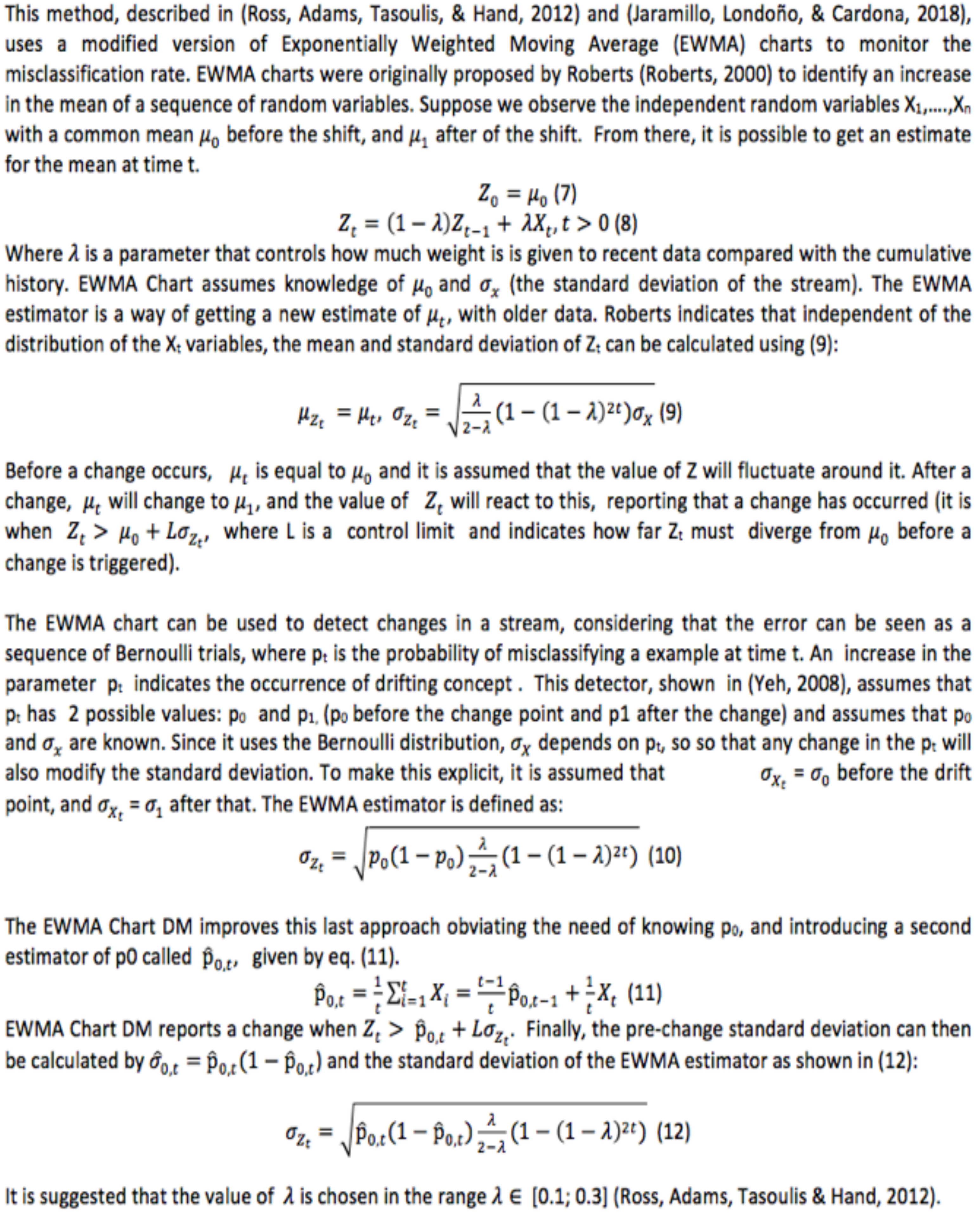
Where $\gamma_i$ is defined as $\gamma_i = \alpha(1-\alpha)^i,\ \ 0 < \alpha \leq 1$ and α is the forgetting factor (α gives more or less weight to the last received data).

The decision function, in a recursive manner, can be rewritten as shown in (6):

$$g_k = (1-\alpha)g_{k-1} + \alpha s_k\ with\ g_0 = 0\ (6)$$

The alarm of drift is defined by the stopping rule $t_\alpha = \min\{k: g_k \geq h\}$, where h is a threshold chosen to tune the sensitivity and false alarm rate of the detector.

## 2.4. Exponentially Weighted Moving Average Chart Detection Method



This method, described in (Ross, Adams, Tasoulis, & Hand, 2012) and (Jaramillo, Londoño, & Cardona, 2018), uses a modified version of Exponentially Weighted Moving Average (EWMA) charts to monitor the misclassification rate. EWMA charts were originally proposed by Roberts (Roberts, 2000) to identify an increase in the mean of a sequence of random variables. Suppose we observe the independent random variables $X_1,....,X_n$ with a common mean $\mu_0$ before the shift, and $\mu_1$ after of the shift. From there, it is possible to get an estimate for the mean at time t.

$$Z_0 = \mu_0 \; (7)$$

$$Z_t = (1-\lambda)Z_{t-1} + \lambda X_t, t > 0 \; (8)$$

Where $\lambda$ is a parameter that controls how much weight is is given to recent data compared with the cumulative history. EWMA Chart assumes knowledge of $\mu_0$ and $\sigma_x$ (the standard deviation of the stream). The EWMA estimator is a way of getting a new estimate of $\mu_t$, with older data. Roberts indicates that independent of the distribution of the $X_t$ variables, the mean and standard deviation of $Z_t$ can be calculated using (9):

$$\mu_{Z_t} = \mu_t, \; \sigma_{Z_t} = \sqrt{\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})\sigma_X} \; (9)$$

Before a change occurs, $\mu_t$ is equal to $\mu_0$ and it is assumed that the value of Z will fluctuate around it. After a change, $\mu_t$ will change to $\mu_1$, and the value of $Z_t$ will react to this, reporting that a change has occurred (it is when $Z_t > \mu_0 + L\sigma_{Z_t}$, where L is a control limit and indicates how far $Z_t$ must diverge from $\mu_0$ before a change is triggered).

The EWMA chart can be used to detect changes in a stream, considering that the error can be seen as a sequence of Bernoulli trials, where $p_t$ is the probability of misclassifying a example at time t. An increase in the parameter $p_t$ indicates the occurrence of drifting concept . This detector, shown in (Yeh, 2008), assumes that $p_t$ has 2 possible values: $p_0$ and $p_1$, ($p_0$ before the change point and p1 after the change) and assumes that $p_0$ and $\sigma_x$ are known. Since it uses the Bernoulli distribution, $\sigma_X$ depends on $p_t$, so so that any change in the $p_t$ will also modify the standard deviation. To make this explicit, it is assumed that $\sigma_{X_t} = \sigma_0$ before the drift point, and $\sigma_{X_t} = \sigma_1$ after that. The EWMA estimator is defined as:

$$\sigma_{Z_t} = \sqrt{p_0(1-p_0)\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})} \; (10)$$

The EWMA Chart DM improves this last approach obviating the need of knowing $p_0$, and introducing a second estimator of p0 called $\hat{p}_{0,t}$, given by eq. (11).

$$\hat{p}_{0,t} = \frac{1}{t}\sum_{i=1}^{t} X_i = \frac{t-1}{t}\hat{p}_{0,t-1} + \frac{1}{t}X_t \; (11)$$

EWMA Chart DM reports a change when $Z_t > \hat{p}_{0,t} + L\sigma_{Z_t}$. Finally, the pre-change standard deviation can then be calculated by $\hat{\sigma}_{0,t} = \hat{p}_{0,t}(1-\hat{p}_{0,t})$ and the standard deviation of the EWMA estimator as shown in (12):

$$\sigma_{Z_t} = \sqrt{\hat{p}_{0,t}(1-\hat{p}_{0,t})\frac{\lambda}{2-\lambda}(1-(1-\lambda)^{2t})} \; (12)$$

It is suggested that the value of $\lambda$ is chosen in the range $\lambda \in [0.1; 0.3]$ (Ross, Adams, Tasoulis & Hand, 2012).

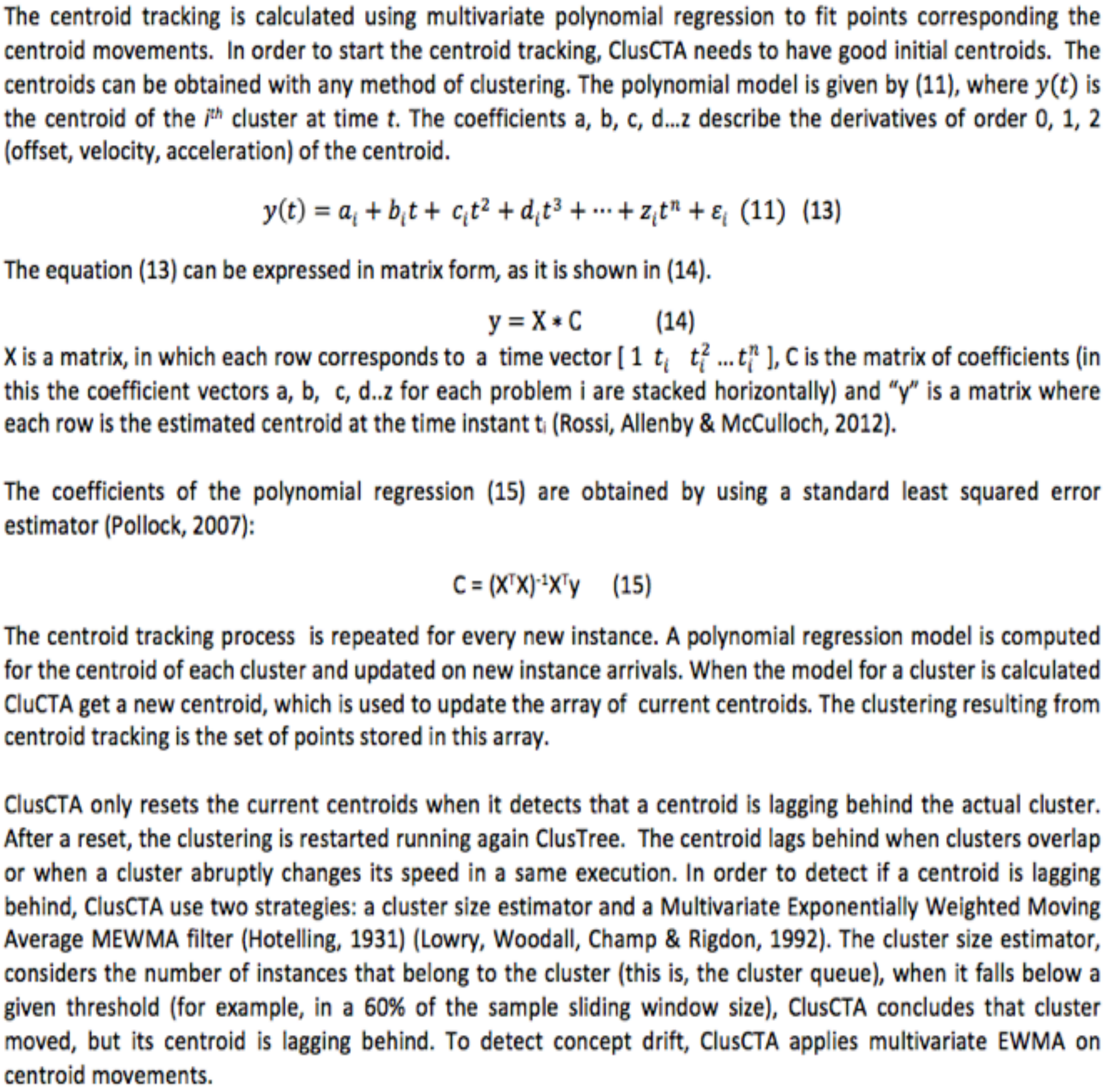
# 3. ClusCTA-MEWMACHART

In this section, we introduce a new technique based on clustering and multivariate EWMA to detect Concept Drift on data streams. The method proposed is referred as ClusCTA-MEWMAChart. ClusCTA-MEWMAChart is a component of ClusCTA. ClusCTA is a new clustering method.

## 3.1. Clustering based on Centroid Tracking (ClusCTA)

Jaramillo et al. proposed ClusCTA. ClusCTA uses multiple sliding windows and centroid tracking for maintaining enough knowledge about centroid behavior. At the beginning, ClusCTA constructs a clustering model by running 5 times the k-Means++ algorithm on the first arrival sample window. It keeps the best of the five models, and based on the estimated radius of the obtained clusters, configures MCOD (Tran, Fan & Shahabi, 2016) (Kontaki, Gounaris, Papadopoulos, Tsichlas & Manolopoulos, 2011) to perform an outlier or noise filtering process. To conclude the initialization phase, ClusCTA invokes ClusTree on the filtered window and it returns the clustering model of the initial sample window. ClusTree (Kranen, Assent, Baldauf & Seidl, 2011) is a compact and self-adaptive index structure, which maintains stream summaries and reports novelty and outliers. The application of the ClusTree on the filtered data allows obtaining a set of good quality initial centroids. It is important for ClusCTA to start from a good quality set of centroids (even if there is noise in the data stream).

The first centroids calculated are stored at an array of current c*centroids.* After this, ClusCTA maintain a sliding window of samples per cluster*with a maximum size*ofn/k*.* When a new instance arrives, it is processed individually, *associated with a cluster (*using a*Nearest Neighbor Criteria)and*recorded in the sliding window of the corresponding cluster. Additionally, ClusCTA have a motion window per cluster (of size m). In this ClusCTA record thearrival time of the new sample and the impact that it has on the movement of its centroid. The computation of the tracking model uses the samples stored in the centroid motion window.



The centroid tracking is calculated using multivariate polynomial regression to fit points corresponding the centroid movements. In order to start the centroid tracking, ClusCTA needs to have good initial centroids. The centroids can be obtained with any method of clustering. The polynomial model is given by (11), where $y(t)$ is the centroid of the $i^{th}$ cluster at time *t*. The coefficients a, b, c, d…z describe the derivatives of order 0, 1, 2 (offset, velocity, acceleration) of the centroid.

$$y(t) = a_i + b_i t + c_i t^2 + d_i t^3 + \cdots + z_i t^n + \varepsilon_i \quad (11) \quad (13)$$

The equation (13) can be expressed in matrix form, as it is shown in (14).

$$y = X * C \qquad (14)$$

X is a matrix, in which each row corresponds to a time vector [ $1 \;\; t_i \;\; t_i^2 \ldots t_i^n$ ], C is the matrix of coefficients (in this the coefficient vectors a, b, c, d..z for each problem i are stacked horizontally) and "y" is a matrix where each row is the estimated centroid at the time instant $t_i$ (Rossi, Allenby & McCulloch, 2012).

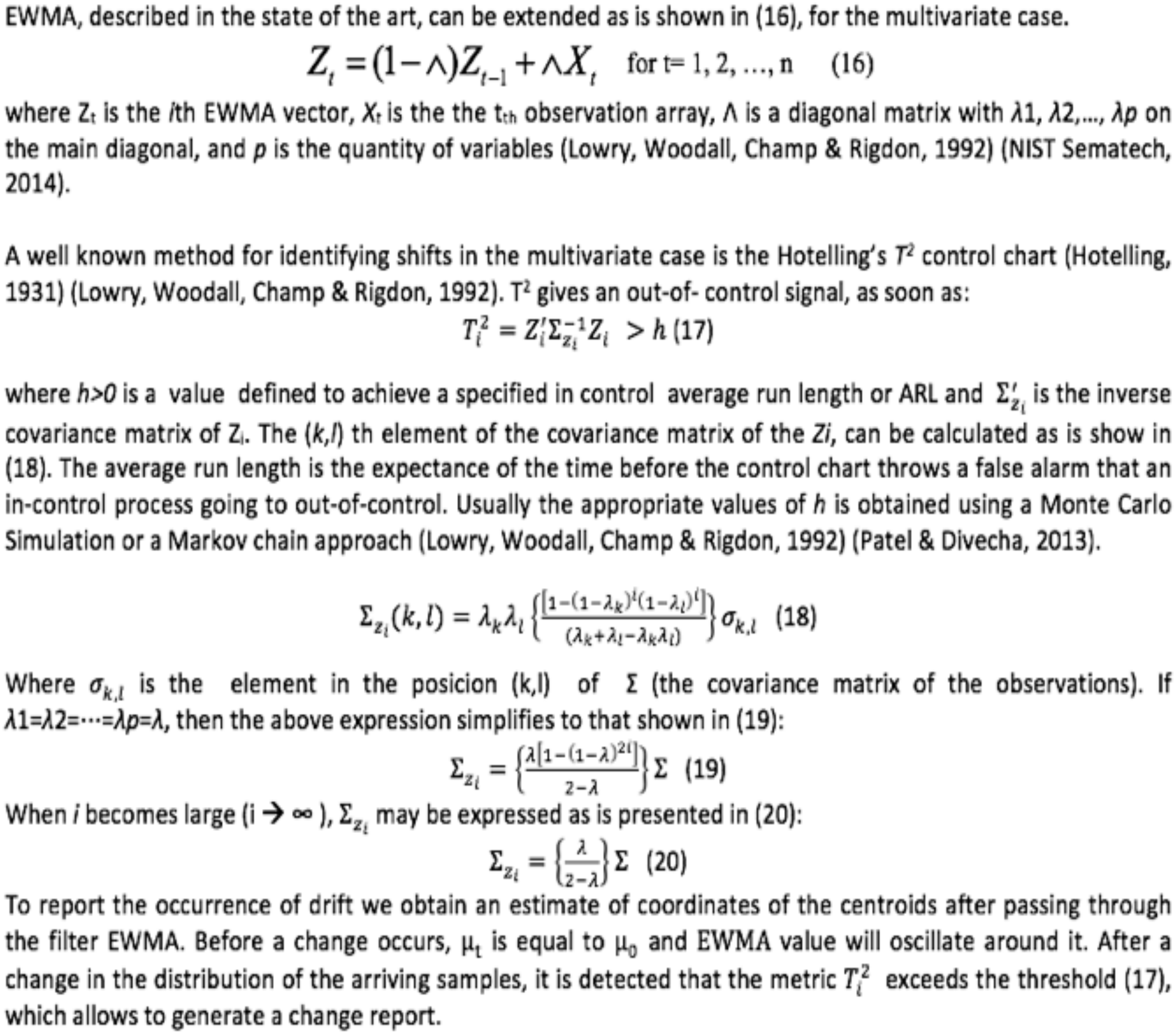
The coefficients of the polynomial regression (15) are obtained by using a standard least squared error estimator (Pollock, 2007):

$$C = (X^T X)^{-1} X^T y \qquad (15)$$

The centroid tracking process is repeated for every new instance. A polynomial regression model is computed for the centroid of each cluster and updated on new instance arrivals. When the model for a cluster is calculated CluCTA get a new centroid, which is used to update the array of current centroids. The clustering resulting from centroid tracking is the set of points stored in this array.

ClusCTA only resets the current centroids when it detects that a centroid is lagging behind the actual cluster. After a reset, the clustering is restarted running again ClusTree. The centroid lags behind when clusters overlap or when a cluster abruptly changes its speed in a same execution. In order to detect if a centroid is lagging behind, ClusCTA use two strategies: a cluster size estimator and a Multivariate Exponentially Weighted Moving Average MEWMA filter (Hotelling, 1931) (Lowry, Woodall, Champ & Rigdon, 1992). The cluster size estimator, considers the number of instances that belong to the cluster (this is, the cluster queue), when it falls below a given threshold (for example, in a 60% of the sample sliding window size), ClusCTA concludes that cluster moved, but its centroid is lagging behind. To detect concept drift, ClusCTA applies multivariate EWMA on centroid movements.

## 3.2. ClusCTA-MEWMAChart



EWMA, described in the state of the art, can be extended as is shown in (16), for the multivariate case.

$$Z_t = (1-\wedge)Z_{t-1} + \wedge X_t \quad \text{for t= 1, 2, …, n} \qquad (16)$$

where $Z_t$ is the *i*th EWMA vector, $X_t$ is the the $t_{th}$ observation array, Λ is a diagonal matrix with *λ*1, *λ*2,..., *λp* on the main diagonal, and *p* is the quantity of variables (Lowry, Woodall, Champ & Rigdon, 1992) (NIST Sematech, 2014).

A well known method for identifying shifts in the multivariate case is the Hotelling's $T^2$ control chart (Hotelling, 1931) (Lowry, Woodall, Champ & Rigdon, 1992). $T^2$ gives an out-of- control signal, as soon as:

$$T_i^2 = Z_i' \Sigma_{z_i}^{-1} Z_i \ > h \ (17)$$

where *h>0* is a value defined to achieve a specified in control average run length or ARL and $\Sigma_{z_i}'$ is the inverse covariance matrix of $Z_i$. The (*k,l*) th element of the covariance matrix of the *Zi*, can be calculated as is show in (18). The average run length is the expectance of the time before the control chart throws a false alarm that an in-control process going to out-of-control. Usually the appropriate values of *h* is obtained using a Monte Carlo Simulation or a Markov chain approach (Lowry, Woodall, Champ & Rigdon, 1992) (Patel & Divecha, 2013).

$$\Sigma_{z_i}(k,l) = \lambda_k \lambda_l \left\{ \frac{\left[1-(1-\lambda_k)^i (1-\lambda_l)^i\right]}{(\lambda_k+\lambda_l-\lambda_k\lambda_l)} \right\} \sigma_{k,l} \quad (18)$$

Where $\sigma_{k,l}$ is the element in the posicion (k,l) of Σ (the covariance matrix of the observations). If *λ*1=*λ*2=⋯=*λp*=*λ*, then the above expression simplifies to that shown in (19):

$$\Sigma_{z_i} = \left\{ \frac{\lambda\left[1-(1-\lambda)^{2i}\right]}{2-\lambda} \right\} \Sigma \quad (19)$$

When *i* becomes large (i → ∞ ), $\Sigma_{z_i}$ may be expressed as is presented in (20):

$$\Sigma_{z_i} = \left\{ \frac{\lambda}{2-\lambda} \right\} \Sigma \quad (20)$$

To report the occurrence of drift we obtain an estimate of coordinates of the centroids after passing through the filter EWMA. Before a change occurs, $\mu_t$ is equal to $\mu_0$ and EWMA value will oscillate around it. After a change in the distribution of the arriving samples, it is detected that the metric $T_i^2$ exceeds the threshold (17), which allows to generate a change report.

# 4. Implementation and method

We implement ClusCTA-MEWMAChart on top of the Massive On-line Analysis MOA framework (The University of Waikato, 2015), a widely known framework for data stream mining written in Java.

The ClusCTA-MEWMAChart estimator was implemented using the Equation (14). The values for $T_i^2$ were obtained by the equation (15). The covariance of the MEWMA vectors was obtained by using the simplified equation, presented in (18).

In ClustCTA-EWMAChart implementation, instead of using Monte Carlo Simulation or a Markov chain approach, as is usual to calculate *h*, we establish a proportion between T and the value of the metric Z that defines the hypervolume that contains 90% of the instances of the cluster (we named it as $d_{pc}$). We throw a drifting alarm, when $T_i$ /$d_{pc}$> h. Since ClusCTA-MEWMAChart can be considered a classifier, which handles two possible classes (with drift or without drift), we use multiobjective optimization to obtain the threshold *h*. We select the *h* which maximizes 2 classification quality indices: accuracy (Rijsbergen, 1979) and Youden's J (Youden, 1950). The accuracy, see (19), is the ratio of true results among the total quantity of examples observed (Metz, 1978). Youden's J, see (20), is a single statistic that corresponds to the best combination of sensitivity and specificity in the prediction and takes values between from -1 to 1 (Youden, 1950).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (19)$$

$$J = \frac{TP}{TP+FN} + \frac{TN}{TN+FP} - 1 \quad (20)$$

In the equations, TP is the number of true positives, FP the number of false positives, TN the number of true negatives and FN the number of false negatives. We identified that 23% is a good percentage to report a drift alert.

For calculate $d_{pc}$, we sort in ascending order the distances between the instances and its centroid and, choose as $d_{pc}$ the distance corresponding to ninth decile. The distance beetwen an instance P and its centroid C is calculated (using Mahalanobis' distance) in terms of Z, as is shown in (21). This metric describes the radius of the hypersphere that contains 90% of the cluster instances, when the $\Sigma_{z_i}^{-1}$ is equal to the identity matrix.

$$d_{pc} = \overline{PC}_i' \Sigma_{z_i}^{-1} \overline{PC}_i \quad (21)$$

where $\overline{PC}$ is the difference *of* the vectors P and C.

When ClusCTA-MEWMAChart reports a drifting event ($T_i$ /$d_{pc}$> h), it resets the clustering model and calculates a new one using the ClusTree algorithm. This also occurs on cluster overlapping events. The lag of centroids causes the previously constructed model to be inconsistent, requiring to be replaced with a new one to prevent accuracy deterioration.

The experiments focus on assessing the accuracy of the methods to correctly identify concept drift in the presence of noise. To test the different algorithms under the same conditions, we generated several synthetic datasets with RandomRBFGeneratorEvents and saved them to files. This way, the same dataset could be used as input to the different methods.

RandomRBFGeneratorEvents is a generator based on the random Radial Basis Function that adds *drift*to samples in a *stream*(Bifet A. , 2012). The random radial basis function, described in (Bifet, Holmes, Pfahringer, & Gavalda, 2009) generates a fixed number of random centroids. Each centroid has a random position, a class label, a standard deviation and a weight. The instances are generated by selecting a centroid at random. For this process, the weights of the centroids are taken into consideration, so centroids with larger weight are more likely to be chosen. The chosen centroid determines the class label of the instance. The random radial basis function gives rise to a normally distributed hyper sphere of instances enclosing each centroid. Drift is added by moving the centroids at a constant rate.

For the evaluation with DDM, EDDM, GeometricMovingAverageDM and EWMAChartDM, we use DriftDetectionMethodClassifier (Baena, 2012), a class for detect concept drift with a wrapper on a classifier. The chosen classifier is a Naive Bayes classifier and the concept Drift Detection Technique can be any of the 4 methods (DDM, EDDM, GeometricMovingAverageDM and

EWMAChartDM. ClusCTA-MEWMAChart does not use DriftDetectionMethodClassifier.

We configure the RandomRBFGeneratorDrift, so that it creates five centroids of which 2 have no movement (speedOption=0) and 3 do have. In a same execution, a cluster with movement can change its speed (speedOption may *take* 3 different *values:* 0.01/500, 0.001/500 and 0.1/500).

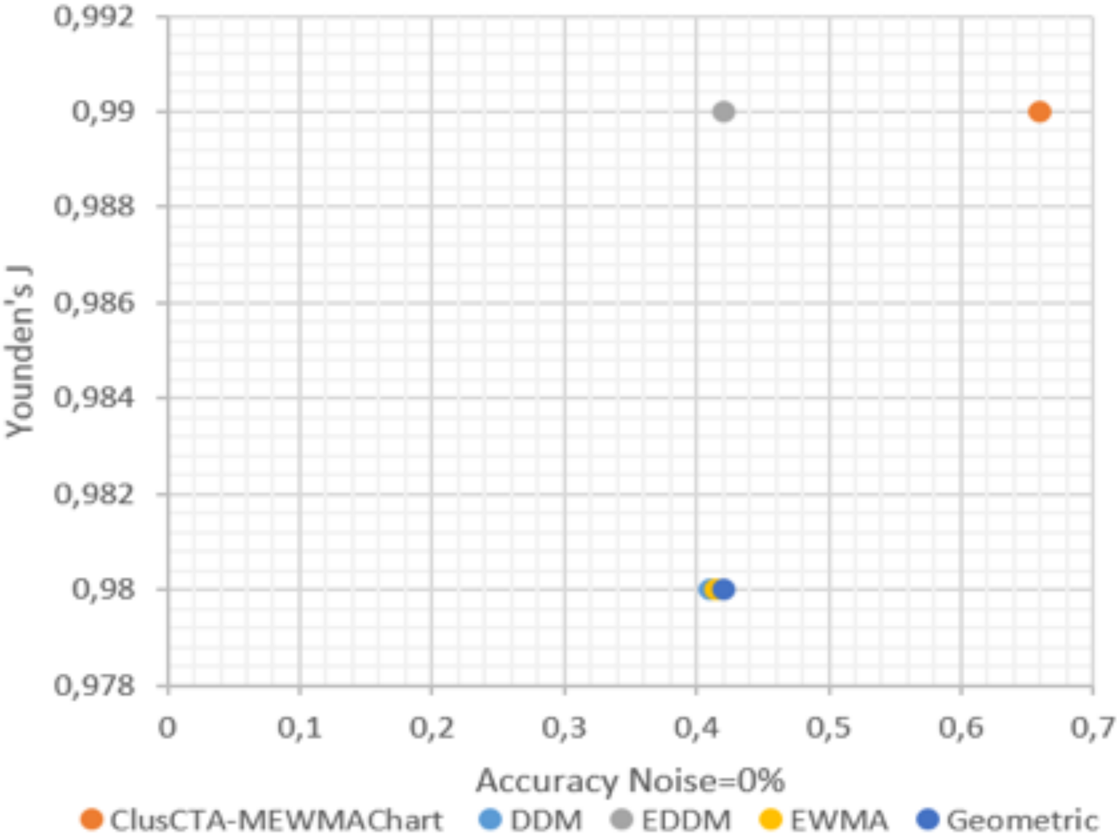
In *data streams,* it is possible to have imbalanced classes, and the accuracy can be misleading when this happens. In this case, it may be desirable to have a model with a lower accuracy but with greater predictive power (Brownlee, 2014). For this reason, to assess the accuracy of the methods to identify concept drift, in addition to accuracy, we use Youden's J statistic.

**Table 1**
The Parameters used for evaluation

| **Concept drift detection method** | **Parameters used for evaluation** |
|---|---|
| **ClusCTA-MEWMAChart** | λ=0.10<br>h =23%<br>WMcovariance=200 (movements of each centroid to calculate the covariance matrix of the Zi per cluster)<br>Wpolynomial=350 (sliding window of 350 samples to calculate the polynomial regression model). |
| **EWMAChartDM** | λ=0.10<br>minNumInstancesOption=30 (minimum number of stored instances before permitting the detection of a change) |
| **DDM** | minNumInstancesOption=30 |
| **EDDM** | minNumInstancesOption=30 |
| **Geometric Moving Average Test** | minNumInstancesOption=30<br>h=1<br>ɑ=0.99 |

To identify the best parameters for each concept drift detection method, we try several values. We use multiobjective optimization to obtain the best parametrization. For this purpose, we select the configuration by maximizing 2 classification quality indices, accuracy and Youden's J. The parameter MinNumInstancesOption did not show influence quality assessment of the methods. The best configuration of each method was selected to compare them against each other method. The results obtained are summarized in Figures 1-5.

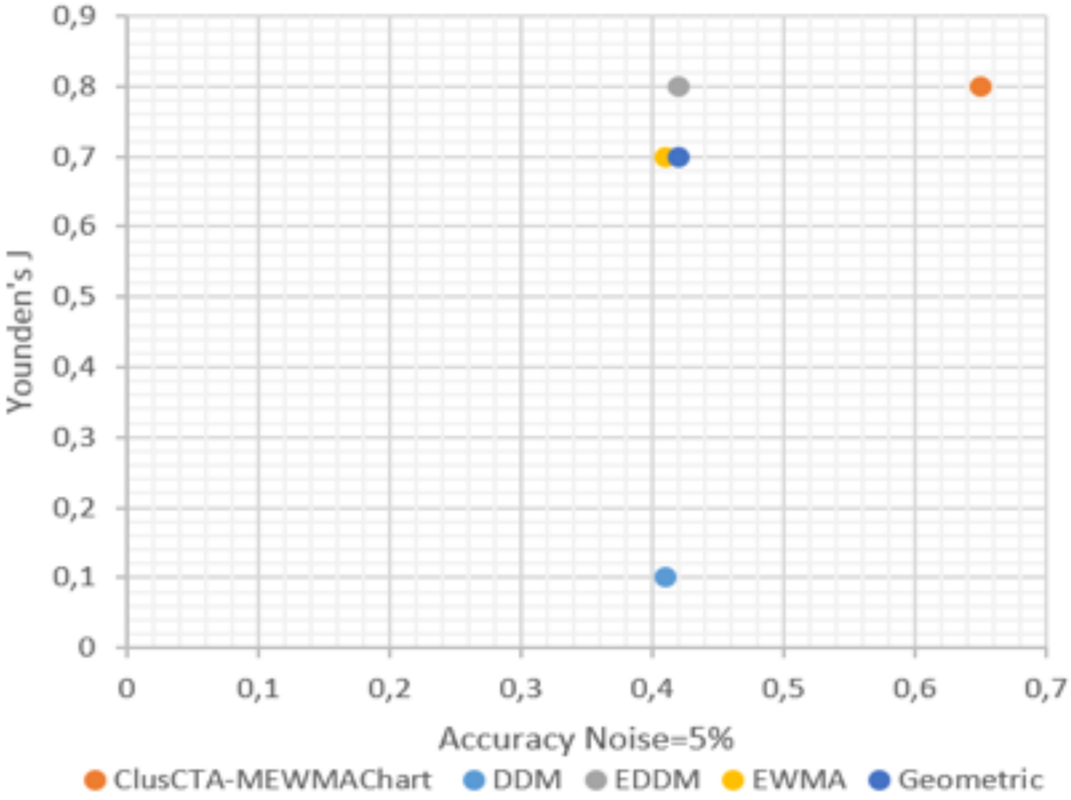
**Figure 1**
Performance comparison for concept drift detection, noise level=0%



*Source: Own image*

-----

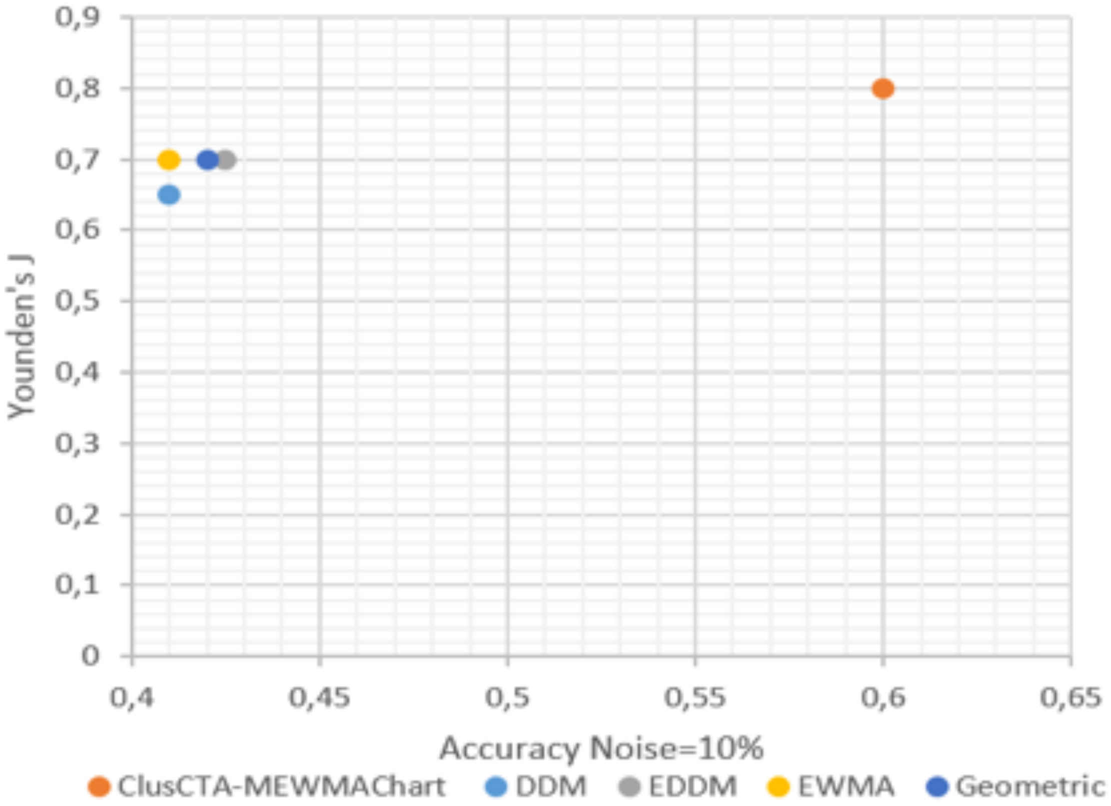
**Figure 2**
Performance comparison for concept drift detection, noise level=5%



*Source: Own image*

-----

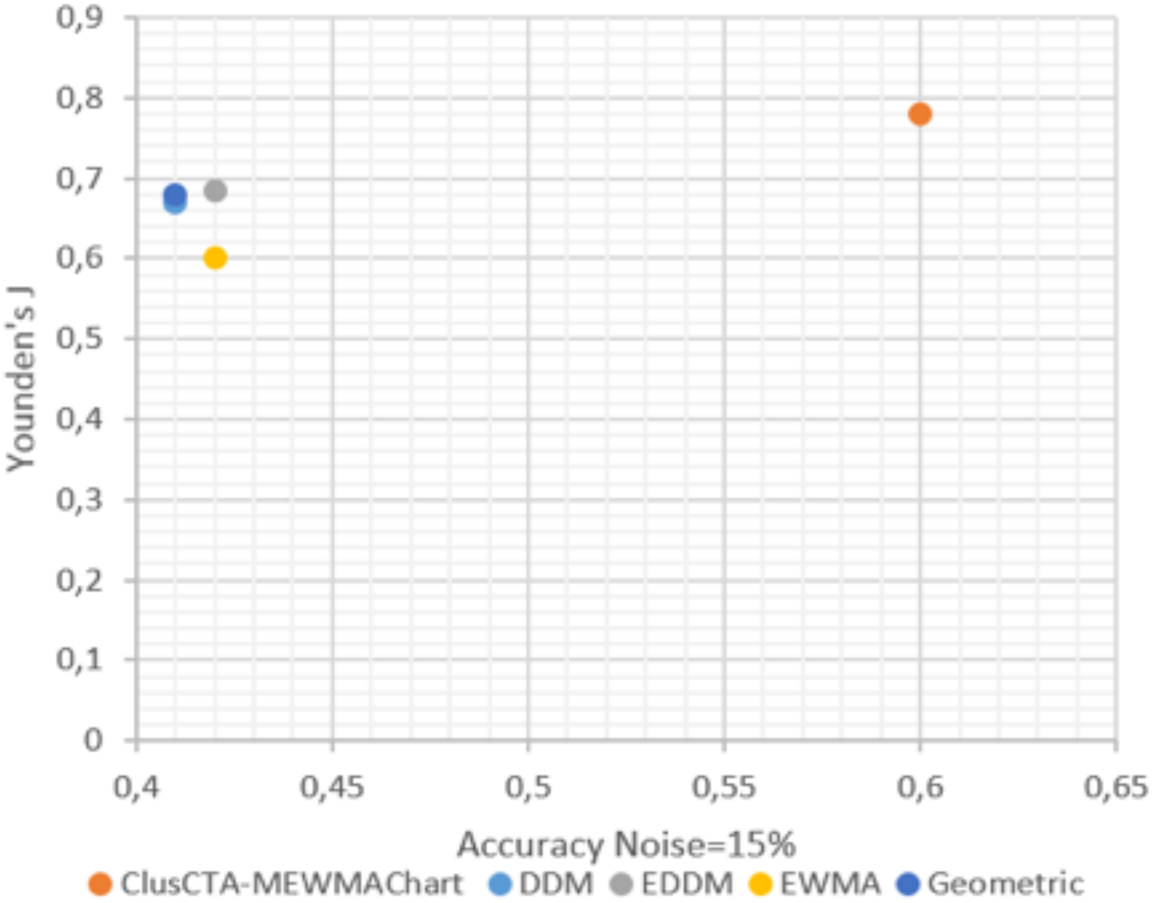
**Figure 3**
Performance comparison for concept drift detection, noise level=10%



*Source: Own image*

-----

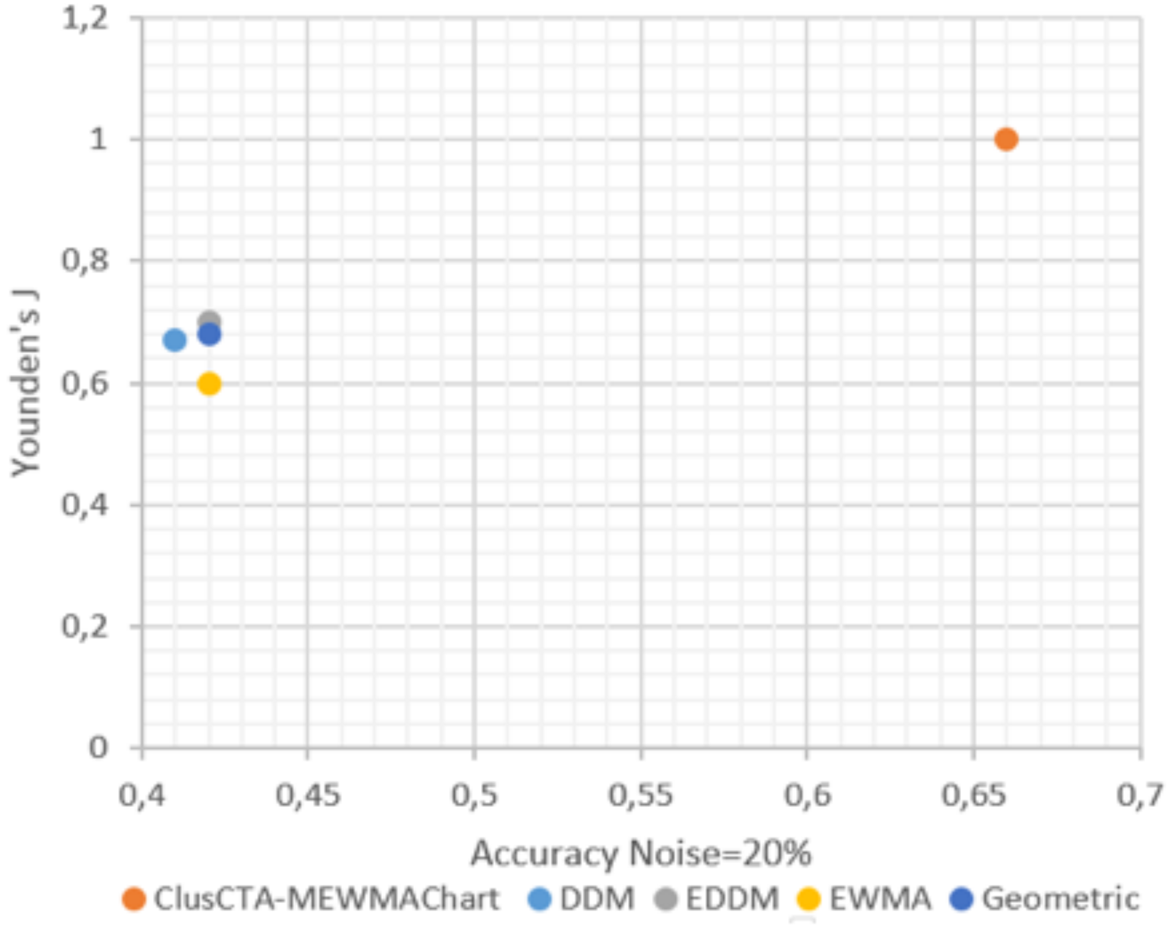
**Figure 4**
Performance comparison for concept drift detection, noise level=15%



*Source: Own image*

-----

**Figure 5**
Performance comparison for concept drift detection, noise level=20%



*Source: Own image*

Figures 1 to 5 report the experimental result for noise level between 0 % and 20%. From the observation, we can find that the classification accuracy and Youden's J is always higher with ClusCTA- MEWMAChart, than with the 4 others methods of drift detection.

When ClusCTA- MEWMAChart detects drift, a new model is induced by applying the clustering algorithm ClusTree in the samples stored.

## 5. Conclusion

In this paper, we present ClusCTA-MEWMAChart as a concept drift detector on centroids, and compare this with other similar approximations with respect to how they performed in the presence of concept drifts, with different velocities of change. ClusCTA-MEWMAChart shows good performance and the centroid tracking proves to be effective to react to concept drift. ClusCTA- MEWMAChart was the best concept drift detection method considering both the accuracy and Youden's J.

ClusCTA-MEWMAChart is designed to detect gradual changes, that is, slow and gradual changes over time. Therefore, how to modify our method to detect abrupt changes is a subject of future research.

Future work includes to extend ClusCTA-MEWMAChart to consider novelty detection (new emergent classes) in data streams scenarios and online class imbalance learning. The process of concept drift detection in imbalanced data is a more complicated task than in the balanced classes, leading to a degradation in performance (Wang, Minku, Ghezzi, Caltabiano & Tino, 2012).

## Bibliographic references

BAENA-GARCIA, M., CAMPO-AVILA, J. D., FIDALGO, R., BIFET, A., GAVALDA, R., & MORALES-BUENO, R. (2006). **Early drift detection method**. Fourth International Workshop on Knowledge Discovery from Data Streams.

BASSEVILLE, M. & NIKIFOROV, I. V. (2012). **Detection of Abrupt Changes**. Retrieved march 2016, from ftp://ftp.irisa.fr/local/as/mb/k11.pdf

CHANDOLA, V., BANERJEE, A., & KUMAR, V. (2009). **Anomaly Detection: A Survey.** ACM Comput. Surv., 41(3), 15:1-15:58. From http://doi.acm.org/10.1145/1541880.1541882

DEL CASTILLO, E. (2001). **Some properties of ewma feedback quality adjustment schemes for drifting disturbances**. Journal of Quality Technology, 33(2).

DONGRE, P. B., & MALIK, L. G. (2014). **Stream Data Classification and Adapting to Gradual Concept Drift**. International Journal of Advance Research in Computer Science and Management Studies, 2 (3).

DONGRE, P., & MALIK, L. (2014, Feb). **A review on real time data stream classification and adapting to various concept drift scenarios.** Advance Computing Conference (IACC), IEEE International, (pp. 533-537).

FANG CHU, A. (2005). **Mining Techniques for Data Streams and Sequences**. University of California at Los Angeles.

GAMA, J. (2010). **Knowledge Discovery from Data Streams**. Chapman & Hall/CRC. Retrieved march of 2016

GAMA, J. M., & RODRIGUES, P. (2004). **Learning with drift detection**. Lecture Notes in Computer Science 3171.

GAMA, J., ZLIOBAITE, I., BIFET, A., PECHENIZKIY, M., & BOUCHACHIA, A. (2014). **A Survey on Concept Drift Adaptation.** (ACM Computing Surveys, Vol. 1, No. 1, Article 1, Publication date: January 2013).

HOTELLING, H. (1931). **The generalization of Student's ratio.** Annals of Mathematical Statistics, 2 (3): 360–378.

JARAMILLO, S., LONDOÑO, J.-M., & CARDONA, S.-A. (s.f.). **Performance Evaluation of Concept Drift Detection Techniques in the presence of noise.** Revista Espacios, Vol.38(38)2017.

KELLY, M., HAND, D. J., ADAMS & M., N. (1999). **The impact of changing populations on classifier performance.** KDD, 367–371.

KONTAKI, M., GOUNARIS, A., PAPADOPOULOS, A. N., TSICHLAS, K., & MANOLOPOULOS, Y. (2011). **Continuous monitoring of distance-based outliers over data streams.** Proceeding ICDE '11 Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, (págs. 135-146).

KRANEN, P., ASSENT, I., BALDAUF, C., & SEIDL, T. (2011). **The ClusTree: Indexing Micro-clusters for Anytime Stream Mining.** Journal Knowledge and Information Systems, 29(2), 249-272. From http://dx.doi.org/10.1007/s10115-010-0342-8

LE, T., STAHL, F., GOMES, J. B., MEDHAT GABER, M., & DI FATTA, G. (2014). **Computationally Efficient Rule-Based Classification for Continuous Streaming Data.** Springer International Publishing Research and Development. Switzerland. doi:10.1007/978-3-319-12069-0_2

LOWRY, C. A., WOODALL, W. H., CHAMP, C. W., & RIGDON, S. E. (1992). **A Multivariate Exponentially Weighted Moving Average Control Chart**. Technometrics, 34(1), 46-53. From http://dx.doi.org/10.2307/1269551

MINKU, L., & YAO, X. (2012). **DDD: A New Ensemble Approach for Dealing with Concept Drift.** Knowledge and Data Engineering, IEEE Transactions on, 24(4), 619-633.

MOUSS, H., MOUSS, D., MOUSS, N., & SEFOUHI, L. **Test of Page-Hinkley, an Approach for Fault Detection in an Agro-Alimentary Production System.** Proceedings of the 5th Asian Control Conference. Vol.2, pages: 815-818, 2004.

NASON, G. P. (2006). **Stationary and non-stationary time series**. H. Mader; S.C Coles. The Geological Society.

NIST SEMATECH. (2014). **Multivariate EWMA Charts**. Retrieved el 19 de oct de 2016, de http://www.itl.nist.gov/div898/handbook/pmc/section3/pmc343.htm

PAGE, E. S. **Continuous Inspection Schemes.** Biometrika, 41:100-115, 1954.

PATEL, A. K., & DIVECHA, J. (2013). **Modified MEWMA Control Scheme for an Analytical Process data**. Global Journal of Computer Science and Technology Software and data Engeneering, 13(3).

POLLOCK, D. (2007). **Regression Analysis.** Retrieved el 02 de december de 2015, de http://www.le.ac.uk/users/dsgp1/COURSES/THIRDMET/MYLECTURES/2MULTIREG.pdf

ROBERTS, S. W. (2000). **Control chart tests based on geometric moving average**. Technometrics, 42(1):97.

ROSS, G. J., ADAMS, N. M., TASOULIS, D. K. & HAND, D. J. (2012). **Exponentially Weighted Moving Average Charts for Detecting Concept Drift**. Pattern Recognition Letters, 33(2) 191-198, 2012.

ROSSI, P. E., ALLENBY, G. M., & MCCULLOCH, R. (2012). **Bayesian Statistics and Marketing**. John Wiley & Sons.

TRAN, L., FAN, L., & SHAHABI, C. (2016). **Distance based Outlier Detection in Data Streams**. Proceedings of the VLDB Endowment, Vol. 9, No. 12.

WANG, H., YU, P., & HAN, J. (2010). **Mining Concept-Drifting Data Streams.** En O. Maimon, & L. Rokach (Edits.), Data Mining and Knowledge Discovery Handbook (págs. 789-802). Springer US.

WANG, H., YU, P., & HAN, J. (2010). **Mining Concept-Drifting Data Streams**. In O. Maimon, & L. Rokach (Eds.), Data Mining and Knowledge Discovery Handbook (pp. 789-802). Springer US. Retrieved from http://dx.doi.org/10.1007/978-0-387-09823-4_40

WIDMER, G., & KUBAT, M. (1996). **Learning in the Presence of Concept Drift and Hidden Contexts**. In d. 10.1007/BF00116900 (Ed.), Journal Machine Learning archive, Volume 23 Issue 1, (pp. 69-101). USA.

YEH, A. M. (2008). **EWMA control charts for monitoring high-yield processes based on non-transformed observations**. International Journal of Production Research 46 (20), 5679-5699.

1. PhD in Engineering. Computer Engineering Dept., Universidad del Quindío, UQ. Armenia, (Colombia), sjaramillo@uniquindio.edu.co

2. PhD in Computer Science. Engineering Dept. Universidad Pontificia Bolivariana, UPB. Medellín, (Colombia), jorge.londono@upb.edu.co

3. PhD candidate in Engineering. Computer Engineering Dept., Universidad del Quindío, UQ, Armenia, (Colombia), sergio_cardona@uniquindio.edu.co