Performance evaluation of concept drift detection techniques in the presence of noise

Evaluación de desempeño de técnicas de detección de concept drift en presencia de ruido

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

Recibido: 26/03/2017 • Aprobado: 23/04/2017

Content
1. Introduction
2. Concept drift detection methods
3. Related works
4. Comparative evaluation
5. Conclusion

Bibliographic references

ABSTRACT:
Many real-world applications generate massive amount of data that are continuous. This kind of data is known as streams. Sensor data, web logs, smart devices, phone records, social networks and ATM transactions are examples of sources of data streams. Typically, data streams evolve over time; this is referred to as concept drift. This phenomenon creates new challenges not present in classical machine learning techniques. In this paper, we compare 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 and real-world datasets. Finally, we present the results and suggest possible directions for future work.

Keywords Concept drift; Classification; Data Stream Mining; adaptive learning

RESUMEN:
Muchas aplicaciones del mundo real generan gran cantidad de datos que son continuos. Este tipo de datos se conoce como streams. Datos de sensores, registros web, dispositivos inteligentes, registros telefónicos, redes sociales y transacciones ATM son ejemplos de fuentes de streams de datos. Tipicamente, los streams de datos evolucionan con el tiempo; esto se conoce como concept drift. El concept drift crea nuevos desafíos que no están presentes en las técnicas clásicas de aprendizaje automático. En este trabajo, se comparan 4 diferentes métodos para detectar concept drift en streams de datos y se determina su robustez en presencia de ruido. Se realizan experimentos sobre datasets sintéticos. Finalmente, se presentan los resultados y se sugieren posibles direcciones para trabajos futuros.

Palabras clave Concept drift; Clasificación; Minería sobre streams de datos; aprendizaje adaptativo

1. Introduction

Big data generates million data per day in streamed manner. Streaming data pose challenging research problems in order to respond to tasks such as statistics maintenance, storage, real time querying and pattern discovery (Gama J.). The real time processing poses important challenges. The classical machine algorithms have to be adapted to analyze huge amounts of streaming data on-the-fly and consider the presence of noise and the evolving nature of the stream (Fang Chu, 2005) (Le, Stahl, Gomes, Medhat Gaber, & Di Fatta, 2014). The latter will be referred to as concept-drift.

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 become less accurate and requiring an update or to be replaced with a new one (Wang, Yu, & Han, Mining Concept-Drifting Data Streams, 2010) (Dongre & Malik, 2014). In (Kelly, Hand, Adams, & M., 1999) indicate that depending on the nature of the concept drift, it can occur in three different ways:

- The prior probabilities of classes $P(c_1),..., P(c_k)$ might change over time.
- Class-conditional probability distributions might change $P(X|c_i)$, $i = 1,..., k$ over time. This kind of concept drift is referred to as virtual drift, because it is possible that $P(X|c_i)$ change without affecting the previous classes (e.g symmetric movement to opposite directions).
- Posterior probabilities $P(c_i|X)$, $i = 1,..., k$ might change. This kind of change is often known as real drift.

An important challenge for Concept Drift Detection algorithms is to distinguish the occurrence of noise (random
deviations or statistical anomalies) in the data from concept drift (Chandola, Banerjee, & Kumar, 2009). There are several approaches designed to identify concept drift. In this paper, we evaluate the performance of 4 of them in the presence of noise. We choose the Drift Detection Method (DDM) (Gama J., Medas, Castillo, & 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).

The paper is organized as follows. Section 2 describes the methods evaluated. Section 3 reviews related works on performance evaluation of concept drift detection techniques in the presence of noise. Section 4 describes the experiments and analysis carried out in this study. The last section provides a summary of the findings in this work.

2. Concept drift detection methods

In this section we describe 4 known algorithms to detect concept drift: DDM, EDDM, GeometricMovingAverageDM and EWMAChartDM.

2.1. Drift detection method (DDM)

The Drift Detection Method (DDM) proposed in (Gama J., Medas, Castillo, & Rodrigues, 2004) 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 (pi) and its standard deviation σi. If the distribution of the samples is stationary, pi 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 pi and for each instance and when pi+si reaches its minimum value, it stores pmin and smin. Then, DDM checks two conditions to detect whether the system is in the warning or drifting level:

- The warning level is activated when pi+si ≥ pmin +2smin. Beyond of this level, it stores the instances anticipating a possible change of context.

- The drift level is triggered when pi+si ≥ pmin +3smin. In this level, DDM resets the variables (pmin and smin) 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 slow but gradual 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)

In (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' + 2s_i' reaches its maximum value. EDDM considers two thresholds:

-Warning level. This level is triggered when (p_i' + 2s_i')/(p_{max}' + 2s_{max}') < α. 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' + 2s_i')/(p_{max}' + 2s_{max}') < β. 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).

2.3. GeometricMovingAverage (GMA) Detection Method
This method, described in (Sadhukha, 2003), 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)

$$E\theta_0(s) < 0 \text{ and } E\theta_1(s) > 0 \quad (2)$$

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).

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

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

The other underlying idea behind GMA refers to the exponential weighting ($y_i$) 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):

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

Where $y_i$ is calculated as $y_i = \alpha(1 - \alpha)^i$, $0 < \alpha \leq 1$ and $\alpha$ is the forgetting factor ($\alpha$ gives more or less weight to the last received data). The decision function, in a recursive manner, can be rewritten as shown in (6):

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

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), 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 $X_1, \ldots, X_n$ 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 $t$.

$$Z_0 = \mu_0$$

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

Where $\lambda$ is a parameter that controls how much weight 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):

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

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).
3. Related works

Performance evaluations of concept drift detection techniques presented in the literature generally consider the prequential error of the classifier and total number of changes detected by each method. In (Gama, Sebastiao, & Rodrigues, Issues in evaluation of stream learning algorithms) define the prequential error as the error obtained using each new instance arrival to test the model before it is used for training. This way the accuracy of the model can be incrementally updated.

Noise and outliers may increase false alarm rates in drift detection algorithms. Few evaluations regarding this problem are available in the literature. Sebastião and Gama (Sebastiao & Gama, 2009) present a study on Change Detection Methods. The experimental evaluation considers 5 different methods for concept drift detection (Statistical Process Control (Gama J., Medas, Castillo, & Rodrigues, 2004), ADaptive WINdowing (Bifet & Gavalda, 2007), Fixed Cumulative Windows Model (Sebastiao & Gama) and Page Hinkley Test (Page) (Mouss, Mouss, Mouss, & Sefouhi)) and 4 fundamental aspects: to be able to detect and react to drift, not to exhibit miss detections, to be resilient to false alarms in stationary environments, and to require few samples to detect a change. The experimental evaluation uses the dataset called SEA Concepts and calculates the error rate (computed using a naive-Bayes classifier), for the different methods. Assessing measures such as: false alarm rates, number of examples until a change is detected, and miss detections rates. This analysis is used to choose the more appropriated algorithms for detecting changes (Page Hinkley Test and the ADaptive WINdowing). The analysis also shows that for the chosen technique there is a trade-off between the rate of false alarms, the miss detections, and the delay until detection. Unlike (Sebastiao & Gama, 2009), our assessment considers the effect of noise on the evaluation of the various methods to determine their robustness in the presence of noisy data.

4. Comparative evaluation

This section describes the evaluation of the different approaches to detect concept drift from data streams described above. These methods are implemented on top of the Massive On-line Analysis (MOA) framework (The University of Waikato, 2015), a widely known framework for data mining evolving data streams written in Java. 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., RandomGeneratorDrift, 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 hypersphere of instances enclosing each centroid. Drift is added by moving the centroids at a constant rate.

For the evaluation, 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 EWMACDM).

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).

For the evaluation, we calculate 2 classification quality indices, accuracy (Rijsbergen, 1979) and Youden's J. The accuracy, see (19), is the ratio of true results among the total quantity of examples observed (Metz, October 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 -1 to 1 (Youden, 1950).

\[
\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (19)
\]

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

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.

Figures 1 to 5 report the experimental results for noise level between 0% and 20%.

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

![Figure 1](Source: Own image)

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

![Figure 2](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%.
From the observation of Figures 1-5, we can find that the classification accuracy and Youden’s J is always higher with EDDM, than with the 3 others methods of drift detection.

DriftDetectionMethodClassifier sends the result of the prediction obtained, with Naive Bayes classifier, to the drift detection approach. If it detects drift, a new model is induced by applying the classification algorithm in the samples stored since the warning level triggered.

5. Conclusion
In this paper, we evaluate 4 different algorithms for concept drift detection in the presence of noise. The experimental results show that the classification accuracy of Naive Bayes classifier is higher with EDDM for 5 levels of noise.

Directions for future work include exploring the 4 approaches in datasets with imbalance of classes and new emergent classes and implementing multivariate EWMA to report the occurrence of drift.

Bibliographic references


