Filomat 32:5 (2018), 1737–1745
https://doi.org/10.2298/FIL1805737S

Published by Faculty of Sciences and Mathematics,
University of Niš, Serbia
Available at: http://www.pmf.ni.ac.rs/filomat

A Fast Learn++.NSE Classification Algorithm Based on Weighted Moving Average

Yan Shen[a,b], Yuquan Zhu[b], Jianguo Du[a], Yong Chen[b]

[a]*Department of Information Management and Information System, Jiangsu University, Zhenjiang, Jiangsu 212013, China*
[b]*School of Computer Science and Communication Engineering, Jiangsu University, Zhenjiang, Jiangsu212013, China*

Abstract. Current researches of incremental classification learning algorithms mainly focus on learning from data in a stationary environment. The incremental learning in a non-stationary environment (NSE), where the underlying data probability distribution changes over time, however, has received much less attentions despite the abundant real applications have generated the long-term and cumulative big data in NSE. Thus, the incremental learning in NSE has gradually received extensive attentions. Nevertheless, the popular incremental classification learning algorithms currently for NSE such as SEA and DWM generally place strict restrictions on the changes. These algorithms can only deal with gradual drift and noncyclical and no new category situations. Therefore, it is highly necessary to develop a novel efficient incremental classification learning algorithm for the gradually cumulative big data in complex NSE. The recently proposed Learn++.NSE algorithm is an important research achievement in this field. However, the vote weight of each base-classifier of the Learn++.NSE depends on its whole error rates in the environments experienced. Therefore, the classification learning efficiency of the Learn++.NSE should be further improved. A novel fast Learn++.NSE algorithm based on weighted moving average (WMA-Learn++.NSE) is presented in this paper, which computes the weighted average of error rates using the sliding window technology to optimize the weight calculation. By only using the recent classification error rates of each base-classifier inside the sliding window to calculate the vote weight, the WMA-Learn++.NSE accelerates the compute of vote weight and improves the efficiency of classification learning. The verification experiments and performance analyses on both synthetic and real data set are presented in this paper. The experimental results show that the WMA-Learn++.NSE can achieve a higher execution efficiency compared to the Learn++.NSE in getting the equivalent classification correct rate.

1. Introduction

With the gradual accumulation of big data, the probability distribution of underlying data always changes, causing the concept change problem of big data mining in a non-stationary environment. It

2010 *Mathematics Subject Classification*. 82A04.
Keywords. Cclassification learning algorithm; Big data mining; Ensemble learning; Incremental learning.
Received: 17 October 2017; Accepted: 7 December 2017
Communicated by Hari M. Srivastava
Research supported by the National Natural Science Foundation of China (61702229,71573107), the Nature Science Foundation of Jiangsu Province of China (BK20150531), the Postdoctoral Science Foundation of Jiangsu Province of China (1401056C), the National Statistical Science Research Project of China (2016LY17), and the Advanced Talent Foundation of Jiangsu University of China (13JDG127)
Email addresses: 104186179@qq.com (Yan Shen), yqzhu@ujs.edu.cn (Yuquan Zhu), jgdu2005@163.com (Jianguo Du), 68151247@qq.com (Yong Chen)

is very common and typical in practice to do incremental classification mining of big data in the non-stationary environment, like spam classification, weather forecast etc. However, there are relatively few researches in this field. To the best of our knowledge, the current algorithms in literature always place strict restrictions on the changes of non-stationary environments. These algorithms can only deal with gradual drift and noncyclical and no new category situations. One of the important targets of classification learning research is to analyze the big data generated by real applications. Since these data sets are often gradually cumulative in the non-stationary environment, it is highly necessary to study the incremental classification algorithm for these big data in such complex NSE. Recently, the proposed Learn++.NSE algorithm emerges as one of the most important achievements in this field [1]. The Learn++.NSE algorithm is essentially a multiple classifier ensemble algorithm that would set up a base-classifier for each data set in batches respectively. At last, the final ensemble classification learning result is got by vote based on the error rates of each base-classifier in all the environments experienced. In lots of synthetic and real data sets, the Learn++.NSE algorithm has achieved remarkable effects, greatly improving the accuracy of classification prediction. However, when calculating the vote weight of each base-classifier, the current Learn++.NSE algorithm needs to take into account all its recent and past classification error rates. In addition, because it is necessary to calculate the voting weights of so many base-classifiers for integrating, the algorithms execution efficiency is in urgent need of being further improved. The recent literature [2, 3] on improving the Learn++.NSE mainly focuses on the process of imbalance data rather than on improving the execution efficiency.

Therefore, in this paper, on the basis of the Learn++.NSE algorithm, a fast Learn++.NSE classification algorithm based on weighted moving average (WMA-Learn++.NSE) is proposed to solve the efficiency problem of incremental classification mining of big data in NSE. Using the sliding window technology after analyzing the characteristics of sigmoid function, the new algorithm considers only the recent classification error rates of each base-classifier to get the vote weight, thus accelerating the weight calculation process of the Learn++.NSE algorithm and improving the classification learning efficiency of the cumulative big data.

2. Review of Existing Approaches

2.1. Relevant algorithms

Due to the appearance of large number of relevant typical application scenarios, more and more attentions have been paid to the research of the incremental classification learning in a non-stationary environment for the cumulative big data. In such an environment, the probability distribution of underlying data is constantly changing. The situation of concept drift and even the concept change would occur. This kind of environment is called non-stationary environment in this paper and can be called NSE for short. Generally, the treatments of the NSE by different algorithms can be divided into: (1) Active detection and passive learning to environment changes. (2) Single classifier and multiple classifiers. (3) Online learning and batch learning.

The active detection algorithms that contain the environment state detection mechanism can identify the changes of environment and then change the learning model. The passive learning algorithms assume the environment is changing all the time and it does not need to be detected. However, the learning model needs to be changed from time to time as the data sets change.

The single classifier algorithms adjust the sole classifier learned from the data to adapt to the changes of environment. The multiple classifiers algorithms gradually establish many base-classifiers in batches according to the changes of environment and then obtain the final classification learning result by integrating these base-classifiers.

The online learning algorithms handle one instance at a time, while the batch algorithms handle a batch of data at a time. The online algorithms have the ability to adapt to the changes of environment more quickly, but have a poor stability. Such algorithms are generally vulnerable to the influence of noise data and the sequence of data. The batch algorithms are more stable than the online learning algorithms but slower in their perception to environmental changes. The quantity of data handled at a time is very important to batch algorithms. If the quantity is too small, the algorithm efficiency is greatly affected. If the quantity is too large, the sensitivity to the environment is decreased. If a batch of data is generated from a variety of environments at the same time, the algorithms are more difficult to handle them. Some of the batch algorithms have utilized the sliding window mechanism, such as the single classifier and passive batch algorithms "STAGGER and FLORA" [4, 5]. These algorithms have set up the dynamic window adjustment mechanism. The size of the window enlarges or shrinks with the different speed of environment change. These algorithms assume that the data outside the window have nothing to do with the current environment, so the information carried by these data can be forgotten. Recently, the batch algorithms are being continuously improved. Some of them try to choose different base-classifiers, such as Decision Tree, Fuzzy Rule, K-nearest neighbor and so on [6–8]. Some of them try to choose different window error thresholds to improve the classification accuracy [9]. However, the main drawbacks of these algorithms lie in that they are not purely incremental and require reading previous data to learn again at a certain time (such as the classification accuracy drops to a preset threshold). Besides, they cannot well deal with periodic environmental changes.

In addition, the newly proposed single classifier and active detection algorithms generally contain a novel detection mechanism for environment changes, such as CUSUM algorithm [10, 11] based on control charts and OLIN algorithm [12] based on the error confidence interval and other methods based on statistical techniques [13]. These algorithms regard the classification learning problem in the non-stationary environment as a kind of prediction problem of environment changes [14] or as the minimization problem of the errors penalty function of classifier [15, 16].

The multiple classifiers ensemble method is an important branch of incremental classification learning algorithms in NSE. Kuncheva divided the multiple classifiers ensemble methods into three categories [17]. (1) Whenever a new batch of data set is learned, a set of fixed classifiers update the combination rules or voting weights [18]. (2) Whenever a new batch of data set is learned, all the existing online classifiers update the parameters [19]. (3) Whenever a new batch of data set is learned, a new classifier is trained using the new data set and is added to the existing integration [1]. In general, these algorithms have a set of base-classifiers with a fixed capacity and use a passive learning mechanism, such as Streets Streaming Ensemble Algorithm (SEA) [20] and Chens Recursive Ensemble Approach (REA) [21]. These algorithms use the newest trained base-classifier to replace the oldest one in the set to construct a new ensemble [22, 23]. Some of the algorithms use the newest trained base-classifier to replace the weakest one in the set to construct a new ensemble, such as the Kolters Dynamic Weighted Majority online algorithm (DWM) [24]. The multiple classifiers ensemble methods mentioned above use their own special vote weight mechanisms.

The Learn++.NSE proposed by Ryan Elwel, et al in 2011 [1] introduces a new voting weight mechanism. It gets the vote weight and the final ensemble result based on each base-classifier's error rates in all the environments experienced. The algorithm is able to deal with a variety of complex environmental changes. In addition, it allows the emergence of new categories. The algorithm is a pure incremental learning algorithm that doesnt need to access the processed data again. Hence, it is particularly useful for the classification learning of cumulative big data, which is not suitable for long-term online storage, such as the video surveillance data and so on. The experimental results of literature [1] prove that the Learn++.NSE algorithm achieves a higher classification accuracy than the single classifier algorithm and the SEA and the DWM. Furthermore, the Learn++.NSE algorithm is able to deal with the complex changes of NSE including the periodical changes that the above algorithms cannot handle very well.

2.2. Learn++.NSE algorithm

The Learn++.NSE algorithm is the latest member of the Learn++ family algorithms. The unique vote mechanism makes this algorithm suitable for the classification learning of gradually cumulative big data from the NSE. Research results of the Learn++.NSE have shown that the algorithm has the following characteristics. (1) Pure incremental learning that means need not access the analyzed data again. (2) A suitable base-classifier can be selected independently according to different applications. (3) A higher classification correct rate can be achieved if the ensemble classifier is not pruned. (4) The classification accuracy is inversely proportional to the speed of environment changes.

The details of the Learn++.NSE algorithm is interpreted below. The input of the Learn++.NSE algorithm includes: the training data set d^t, which is to be processed in the $t(th)$ time. d^t is composed of the instances

$x^t(i)$ and the size of the $x^t(i)$ is $m^t, i = 1, ..., m^t$. A base classifier is required in order to construct the final ensemble result, which is a strong classifier. Parameters a and b are used to adjust the shape of the sigmoid weight function.

When $t = 1$, the instance weight w_i^1 and penalty weight D_i^1 are set to be equal weights, $D_i^1 = w_i^1 = 1/m^t, \forall i$. In the subsequent $t(th)$ time point, parameters w_i^t and D_i^t are determined according to the classification accuracy of the ensemble classifier on the new data set. A new base-classifier h_t^t would be trained on the new data set at $t(th)$ time point. Then, all the existing base-classifiers require computing the classification error rates on the new data set, $\varepsilon_k^t = \sum_{i=1}^{m^t} D_i^t[[h_k^t(x^t(i) \neq y^t(i))]]$ for $k = 1, ..., t$. The error rate is computed through the penalty weight D_i^t, $D_i^t = \omega_i^t / \sum_{i=1}^{m^t} \omega_i^t$. The process method of base-classifiers with different error rates in different periods is distinct. For the current generated base-classifier, if the error rate $\varepsilon_{k=t}^t$ is greater than 0.5, then the base-classifier is invalid and a new base-classifier is to be learned to replace the invalid one. For the base-classifier generated previously at time point $k, k < t$, if the error rate $\varepsilon_{k<t}^t$ is greater than 0.5, then its error rate $\varepsilon_{k<t}^t$ is set to be 0.5. h_k^t indicates that the base-classifier generated at time point k is used at time point t. At last, all the base-classifiers are integrated with different weights to construct the final ensemble classifier $H^t(x^t(i)) = \arg\max_c \sum_k W_k^t \cdot [[h_k^t(x^t(i) = c)]]$. The vote weight of each base-classifier is determined by its weighted average error rate $\overline{\beta_k^t}$, $W_k^t = \lg(1/\overline{\beta_k^t})$. In order to reduce the fluctuation of the average error rate, the average error rate $\overline{\beta_k^t}$ is computed through the weighted average of all normalized error rates of base-classifier in the experienced environments using the sigmoid weight function, $\omega_k^n = 1/(1 + e^{-a(-t+n+b)}), n = k, ..., t, (\omega_k^n)' = \omega_k^n / \sum_{n=k}^{n=t} \omega_k^n, \overline{\beta_k^t} = \sum_{j=0}^{t-k} (\omega_k^{t-j})' \beta_k^{t-j}, fork = 1, ..., t$. This method can let the base-classifier with a lower average error rate so far have a larger vote weight and play a more significant role in the final ensemble result.

The experimental results show that, compared to the single classifier algorithm and SEA and DWM algorithm, the Learn++.NSE algorithm improves the classification correct rate a lot. However, the classification learning process of the Learn++.NSE is relatively slow. Its execution efficiency still needs to be further improved especially when mining the big data. The analysis of its time complexity reveals that the most time-consuming phases mainly include: (1) Training the base-classifier for each batch of data. (2) Computing the average error rate $\overline{\beta_k^t}$ for each base-classifier. (3) Integrating multiple base-classifiers to obtain the final ensemble result $H^t(x^t(i)) = \arg\max_c \sum_k W_k^t \cdot [[h_k^t(x^t(i) = c)]]$. For phase 1, the training time depends on the data characteristics and the choice of base-classifier. For phase 3, the integrated base-classifiers can be pruned. The research results of literature [1] have proved that directly pruning classifiers based on timeliness or error rate will reduce the classification correct rate of the ensemble classifier. Therefore, the current researches mainly aim at phase 2. This paper argues that it is not necessary to consider all the error rates but only the recent error rates, when calculating the weighted average error rate of each base-classifier. A fast Learn++.NSE classification algorithm based on the weighted moving average (WMA-Learn++.NSE) is proposed to speed up the calculation of vote weights when base-classifiers are integrated, which improves the classification learning efficiency of the Learn++.NSE.

3. A Fast Learn++.NSE Classification Algorithm Based on Weighted Moving Average

First of all, when dealing with the new data set, the Learn++.NSE algorithm needs to calculate the error rate E^t of the current ensemble classifier on the new data set. According to the error rate E^t, the penalty weights D_i^t can be obtained, which are required to calculate the error rate ε_k^t of each base-classifier. However, when $E^t = 0$, the process method is not explained in literature [1]. It is worth noting that when $E^t = 0$, according to step 2 of the original algorithm flow, the divisor is 0 now, D_i^t cannot be calculated, $\omega_i^t = \frac{1}{m^t}\begin{cases} E^t, & H^{t-1}(x^t(i) = y^t(i)) \\ 1, & otherwise \end{cases}$ $, D_i^t = \omega_i^t / \sum_{i=1}^{m^t} \omega_i^t$. It is revised by the WMA-Learn++.NSE algorithm here. When When $E^t = 0$, it indicates that the current ensemble classifier can classify the new data set completely right and the current ensemble classifier has completely contained the information of the data set so far. Therefore, the WMA-Learn++.NSE algorithm does not do any incremental learning for the new data set at this moment. That means there is no new base-classifier generated and the ensemble classifier remains unchanged. Only when $E^t \neq 0$, would WMA-Learn++.NSE do incremental learning.

Secondly, when calculating the vote weight of each base-classifier, the Learn++.NSE algorithm needs to compute the average error rate of each base-classifier. In the calculation of the average error rate, the Learn++.NSE algorithm must consider each base-classifiers current and past error rates in all the environments experienced. Additionally, in order to prevent the fluctuation of the average error rate, the error rate is given a weighting using the sigmoid function, $\omega_k^n = 1/(1 + e^{-a(-t+n+b)}), n = k, ..., t, (\omega_k^n)' = \omega_k^n / \sum_{n=k}^{n=t} \omega_k^n$. This method can make the recent classification performance of the base-classifier play a more significant role. Take a weighted calculation process of a real experiment for example, as illustrated in Fig.1. Here the parameters are $a = 0.5$ and $b = 20$. It can be found that the sigmoid function declines very fast on the left and the left side of it are almost zero. Therefore, it is not very useful to distinguish the different performances of base-classifiers when giving a weighting to the error rates of each base-classifier utilizing the values in this interval of sigmoid function. Instead, it would increase the computation complexity of vote weight and reduce the execution efficiency of the Learn++.NSE.

In fact, when calculating the average error rate of each base-classifier, it will be more efficient to only consider the recent classification error rates of it. Furthermore, the recent performances of the base-classifier can better predict its performance in the immediate following environment. After analyzing the shape of the sigmoid function, it can be found that its interval with larger weights is mainly concentrated in $[t-2b, t]$ when fixing the parameter a. Therefore, the sliding window is set to be $2b$. The weighted average error rate of each base-classifier is obtained by giving a weighting using the value between $[t - 2b, t]$ of the sigmoid function after moving the window to the time point t. Therefore, step 5 of the original algorithm can be modified to $\omega_k^n = 1/(1 + e^{-a(-t+n+b)}), n = \max[t - 2b, k], ..., t, (\omega_k^n)' = \omega_k^n / \sum_n \omega_k^n, \overline{\beta_k^t} = \sum_n (\omega_k^n)' \beta_k^n$. The average error rates is calculated by giving a weighting to the error rates of each base-classier inside the sliding window. This method can speed up the weight calculation of each base-classifier.

To optimize the Learn++.NSE's execution efficiency, a fast Learn++.NSE classification algorithm based on the weighted moving average (WMA-Learn++.NSE) is presented in this paper. On the basis of the Learn++.NSE, the calculation process of weighted average of base-classifiers' error rates is accelerated by integrating the sliding window technology. This reduces the computation complexity of average error rate of each base-classifier and improves the efficiency of ensemble learning. This novel method makes the WMA-Learn++.NSE more suitable for classification learning aiming at the accumulated big data over a long period of time.

4. Experimental results and analysis

The performance of the WMA-Learn++.NSE proposed was tested and analyzed by the data set used in literature [1]. The WMA-Learn++.NSE was compared with the Learn ++ NSE algorithm on the classification accuracy and the training time respectively.

The experimental environment included the ThinkPad 20FWA00VCD notebook, the Core i7-6700@2.6GHz CPU, 8GB memory capacity and the Win10 64bit OS. All experiments followed the following settings. (1) The experiment started at time point $t = 0$ nd stopped at $t = 1$ logically. The base-classifiers were trained by the T batches data set gradually arrived at during this period. These T batches of data set were generated by the non-stationary environments. The characteristics of the environments and the distributions changes of the data were all unknown in advance. The value of T represents the change speed of the environment. The larger value of T represents the relatively slow environmental change, while the smaller value of T represents the great environmental change. Because the smaller T value is, the longer interval is when the base-classifier is trained after a batch of data set arrives. (2) During the experiments, the WMA-Learn++.NSE and the Learn++.NSE used the same parameters $a = 0.5, b = 20$ and also used the same base-classifier, which is SVM. SVM was implemented by Weka API. All the parameters of the base classifier SVM were the same. The kernel function was PolyKernel $k(x, y) =< x, y > \wedge p$. The exponential was 1 and the cut-off error was $\epsilon = 1.0 \times 10^{-12}$ and the other parameters used the default values of Weka. (3) Each algorithm carried out 50 independent experiments and recorded the classification accuracy at each time point and the entire running time to calculate the average value. A single classifier-SVM algorithm also included for comparative analysis. The comparisons are very important. Because the single classifier is trained by the latest batch of data without using an ensemble method, this kind of algorithm can track the changes of environment more quickly. This comparison can find out, in a non-stationary environment, whether the Learn++.NSE family algorithms based on learned knowledge, which is contained in the previous base-classifiers, can benefit from the ensemble learning.

4.1. Rotating Checkerboard Dataset

A non-Gaussian data set representing a rotating checkerboard was used in this experiment, as shown in literature [1]. It was a challenging classification problem owing to the increase of the different rotating speeds in this experiment. This was a classification learning problem under the non-stationary periodic environment. The classification scene appeared repeatedly at every π rotating angle. In order to increase the diversity and complexity of classification learning process, 10% noise data were added to the training data set. Each batch of data was composed of 25 instances extracted from the sampling window and each test data set was composed of 1024 instances with the true class labels in the corresponding sampling window by a resolution of $32 * 32$, which is the same as literature [1].

The most challenging part of this classification learning problem was the change in the rotating speed of the checkerboard. The T value was set to be 400 in all the experiments. That means there were 400 equal time steps from time 0 to time 1 and 400 batches of data were processed continuously. The checkerboard rotated 2π radians totally with different rotating speeds. Three different rotating speeds were tested respectively. (1) Constant speed. The checkerboard rotated $2\pi/400$ radians constantly. (2) Sinusoidal speed. The rotating speed conformed to the sinusoidal function. (3) Pulse speed. The rotating speed conformed to the pulse Gaussian function. The function curves of the checkerboard rotating speed were shown in Fig.2.

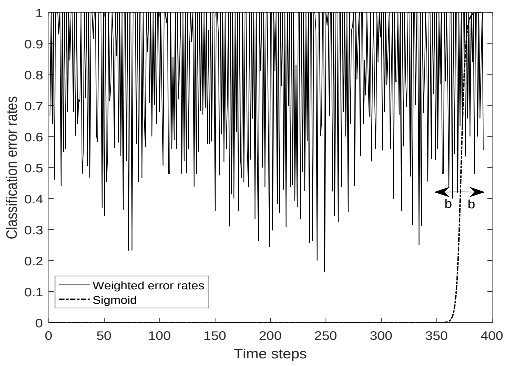

Figure 1: Weighted error rate based on sigmoid function

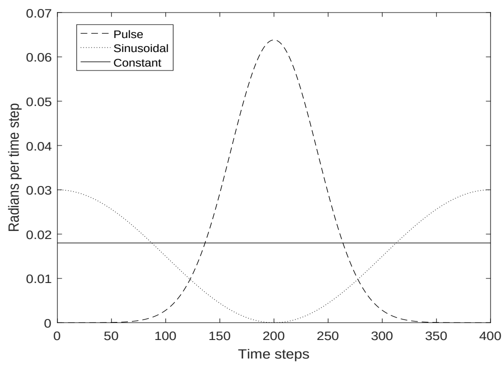

Figure 2: Rotate rate of checkerboard

It can be concluded from the experimental results from Fig.3 to Fig.5 that the Learn++.NSE and the WMA-Learn++.NSE algorithm, which use multiple classifiers ensemble method, both achieved a higher classification accuracy compared to the single classifier algorithm. This conclusion also can be supported by Table 1(a). In addition, it can be found that the accuracy of the ensemble classifier was inversely proportional to the rotating speed of the checkerboard. For example, in the middle part of the sinusoidal rotating and both ends of the pulse rotating, when the checkerboard both rotated slowly, the Learn++.NSE and the WMA-Learn++.NSE algorithm both achieved a higher classification accuracy. Besides, it can be found that the classification accuracy of the ensemble classifier in the second part of the learning scene (π~2π) was better than that in the first half (0~π). This result proved that the method of multiple classifiers integrating can benefit from the classification knowledge contained in the previous classifiers and can better handle the scene reproduction and periodic environment. When observing the classification accuracy of the Learn++.NSE and the WMA-Learn++.NSE at different time points from Fig.3 to Fig.5, it can be found that the classification accuracy of the two algorithms were basically the same. It can also be concluded from

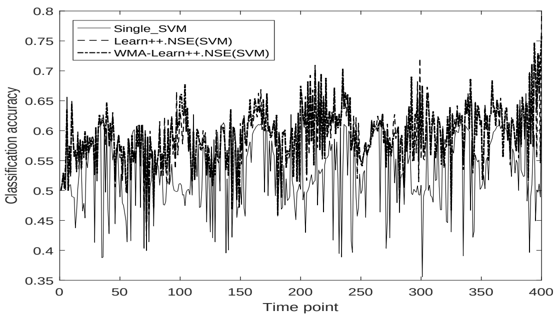

Figure 3: Classification correct rate of constant

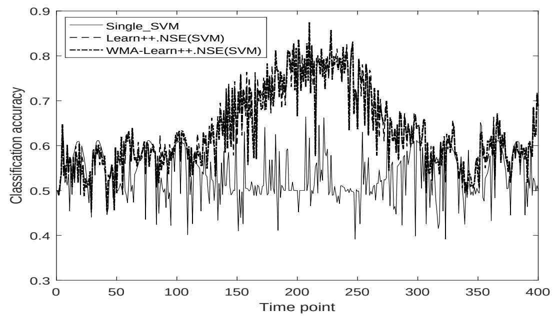

Figure 4: Classification correct rate of sinusoidal

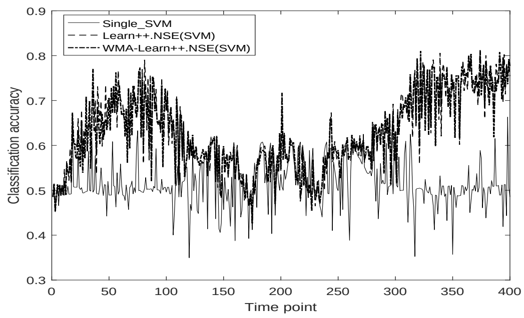

Figure 5: Classification correct rate of pulse

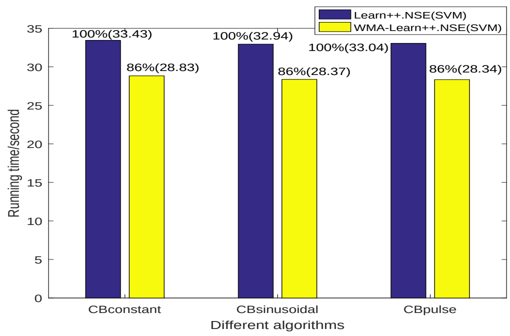

Figure 6: Compare and analysis of running time (CB)

Table 1(a) that there was no significant difference on the classification accuracy between the two algorithms. In contrast, it can be concluded from Fig.6 that when getting the equal classification accuracy, the WMA-Learn++.NSE can improve the execution efficiency a lot compared to the Learn++.NSE. The experimental results had shown the optimization effects of the WMA-Learn++.NSE.

4.2. *Weather forecast data set*

The U.S. National Oceanic and Atmospheric Administration has compiled and released a partial weather data set from over 9000 weather stations worldwide. These weather data records can be dated back to 1930, providing long-term and wide weather trends. Each weather record contains many attributes, such as temperature, air pressure and wind speed etc. Besides, each weather record contains a category tag to indicate whether it rains. This is a real world data set to test the classification performance of different algorithms. It is different to the data sets above, which were generated artificially. The weather data set in 50 years (1949~1999) of the Offutt Air Force Base in Bellevue, Nebraska was chosen from the entire data set to test the classification performance of different algorithms. This is a long-term and gradually accumulated big data set. 8 attributes of it were selected according to the attribute availability and the remaining attributes with a loss rate more than 15% were not selected. The data set contains 18159 records, in which 5698(31%) records classification label is rain and the other 12461(69%) records classification label is no rain. Each time the first 30 records were selected for training and the subsequent 30 records for testing, which formed totally 605 batches of data set in different time points to execute the incremental classification learning. The experimental results are shown in Fig.7 and Fig.8. It can be found from Fig.7 that the classification accuracy of the three algorithms all showed a strong periodicity, which revealed the natural annual cycle characteristics of the weather data. Among the three algorithms, the Learn++.NSE algorithm and the WMA-Learn++.NSE algorithm both greatly improved the classification accuracy compared to the single classifier algorithm (Table 1(b)). The promotion degree was more significant especially when the single classifier's classification accuracy was low.

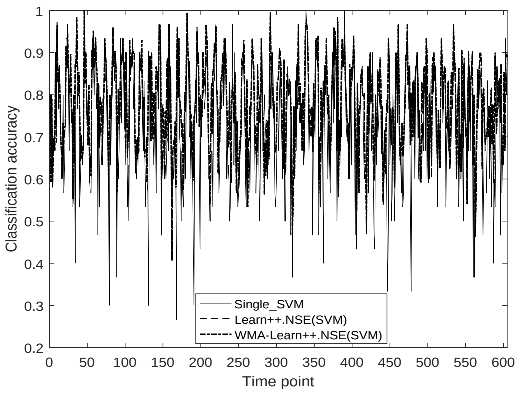

Figure 7: Weather predict data set of 50 years

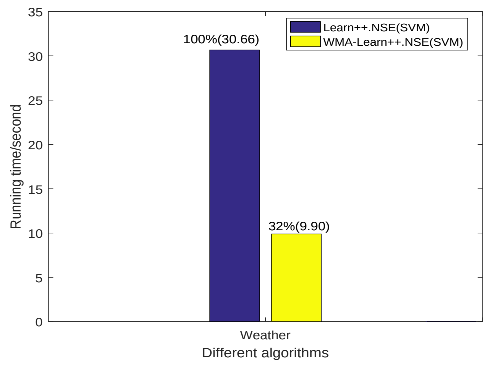

Figure 8: Compare and analysis of running time (Weather)

Table 1: The 95% confident interval of average accuracy rates of the three algorithms

(a) Checkerboard data set

	CB(Constant)	CB(Pulse)	CB(Sin)
Learn++.NSE	0.591 ± (0.002)	0.627 ± (0.003)	0.633 ± (0.003)
WMA-Learn++.NSE	0.592 ± (0.002)	0.627 ± (0.003)	0.635 ± (0.003)
Single_SVM	0.532 ± (0.006)	0.516 ± (0.005)	0.531 ± (0.005)

(b)Weather data set

	Weather
Learn++.NSE	0.777 ± (0.003)
WMA-Learn++.NSE	0.778 ± (0.003)
Single_SVM	0.724 ± (0.011)

It can be found from the different running times of the Learn++.NSE and the WMA-Learn++.NSE in Fig.8 that the WMA-Learn++.NSE algorithm greatly improved the execution efficiency compared to the Learn++.NSE on the premise of getting the equivalent classification accuracy (Table 1(b)). Compared with the experimental results on the previous data sets, this experiment can better show the performance improvement by the WMA-Learn++.NSE. Because the weather forecast data set contains 605 batches of data in succession and this is a long-term accumulated big data set. For this data set, the Learn++.NSE must consider the error rates of each base-classifier in all experienced environments (605 different situations at most). However, the WMA-Learn++.NSE only needed to consider the error rates falling within the sliding window whose size was $2b$ at most ($2 * 20 = 40$ different situations at most in this experiment), thus greatly improving the execution efficiency.

It can be inferred from Table 1 that the Learn++.NSE and the WMA-Learn++.NSE can get a higher classification correct rate compared to the single SVM significantly. In addition, there is no significant difference between the Learn++.NSE and the WMA-Learn++.NSE on the classification accuracy. However, according to the experimental results above, it can be found that the WMA-Learn++.NSE can improve the classification learning efficiency a lot through a sliding window technology to compute the weighted moving average, which is compared to the Learn++.NSE.

5. Conclusion

On the basis of the Learn++.NSE algorithm, a fast Learn++.NSE classification algorithm based on the weighted moving average (WMA-Learn++.NSE) is proposed in this paper by using the sliding window technique. It can improve the efficiency of classification learning for big data accumulated gradually and generated by batch. Theoretical analysis and experimental results based on both artificially generated and real world data sets have shown that the WMA-Learn++.NSE algorithm is more efficient for long-term and accumulated big data under the premise of obtaining an equivalent classification accuracy compared with the Learn++.NSE algorithm.

References

[1] Elwell R, Polikar R, Incremental Learning of Concept Drift in Nonstationary Environments, IEEE Transactions on Neural Networks, 22(10)(2011) 1517–1531.
[2] G.Ditzler, R. Polikar, Incremental Learning of Concept Drift from Streaming Imbalanced Data, IEEE Transactions on Knowledge and Data Engineering, 25(10)(2013) 2283–2301.
[3] Yin X C, Huang K, Hao H W, DE 2: Dynamic ensemble of ensembles for learning nonstationary data, Neurocomputing, 165(2015) 14–22.
[4] Lemaire V, Salperwyck C, Bondu A, A Survey on Supervised Classification on Data Streams,Springer International Publishing, New York, 2015.
[5] Song G, Ye Y, Zhang H, et al, Dynamic Clustering Forest: An ensemble framework to efficiently classify textual data stream with concept drift, Information Sciences, 357(2016) 125–143.
[6] Sheu J J, Chu K T, Li N F, et al, An efficient incremental learning mechanism for tracking concept drift in spam filtering, Plos One, 12(2)(2017) 1–17.
[7] He Runxin, Gonzalez Humberto, Zoned HVAC Control via PDE-Constrained Optimization, Proc of the American Control Conference, 7(2016) 587–592.
[8] Gomes H M, Barddal J P, Enembreck F, Pairwise combination of classifiers for ensemble learning on data streams, Proc of the ACM Symposium on Applied Computing, (2015) 941–946
[9] Wang S, Minku L L, Ghezzi D, et al, Concept drift detection for online class imbalance learning, Proc of the International Joint Conference on Neural Networks, (2013) 1–10.
[10] Alippi C, Boracchi G, Roveri M, Just-in-time classifiers for recurrent concepts, Neural Networks and Learning Systems IEEE Transactions, 24(4)(2013) 620–634.
[11] Sojodishijani O, Ramli A R, Just-in-time adaptive similarity component analysis in nonstationary environments, Journal of Intelligent and Fuzzy Systems Applications in Engineering and Technology, 26(4)(2014) 1745–1758.
[12] Cohen L, Avrahami-Bakish G, Last M, et al, Real-time data mining of non-stationary data streams from sensor networks, Information Fusion, 9(3)(2008) 344–353.
[13] Turkov P, Krasotkina O, Mottl V, Dynamic Programming for Bayesian Logistic Regression Learning under Concept Drift, Lecture Notes in Computer Science, 8251(2013) 190–195.
[14] Rutkowski L, Pietruczuk L, Duda P, et al, Decision Trees for Mining Data Streams Based on the McDiarmid's Bound, IEEE Transactions on Knowledge and Data Engineering, 25(6)(2013) 1272–1279.
[15] Karim M R, Farid D M, An adaptive ensemble classifier for mining complex noisy instances in data streams, International Conference on Informatics and Electronics and Vision, (2014) 1–4.
[16] Ditzler G, Roveri M, Alippi C, et al, Learning in Nonstationary Environments: A Survey, IEEE Computational Intelligence Magazine, 10(4)(2015) 12–25.
[17] Dez-Pastor J F, Rodrguez J J, Garca-Osorio C, et al, Random Balance: Ensembles of variable priors classifiers for imbalanced data, Knowledge-Based Systems, 85(2015) 96-111.
[18] Jiang Y, Zhao Q, Lu Y, Ensemble based Data Stream Mining with Recalling and Forgetting Mechanisms, Proc of the 2014 10th international conference on natural computation, (2014) 430–435.
[19] Kanoun K, Schaar M V D, Big-data streaming applications scheduling with online learning and concept drift detection, Design, Automation and Test in Europe Conference and Exhibition, (2015) 1547–1550.
[20] Senthamilarasu S, Hemalatha M, Ensemble Classifier for Concept Drift Data Stream, Springer, India, 2015
[21] A. Jain P, Raut R, R. Deshmukh P, Recursive Ensemble Approach for Incremental Learning of Non-Stationary Imbalanced Data, International Journal of Computer Applications, 98(17)(2014) 41–45.
[22] Kim K, Lin H, Jin Y C, et al, A design framework for hierarchical ensemble of multiple feature extractors and multiple classifiers, Pattern Recognition, 52(C)(2016) 1–16.
[23] Zhang Zhongping, Jia Qiannan, Deng Changshou, Differential evolution algorithm based on classification and cross-combination, Proc of the 2015 International Conference on Fuzzy System and Data Mining FSDM 2015, (2015) 310–316.
[24] Kolter J Z, Maloof M A, Dynamic Weighted Majority: An Ensemble Method for Drifting Concepts, Journal of Machine Learning Research, 8(12)(2007) 2755–2790.