Detecting Concept Drift Using Statistical Testing*

Kyosuke Nishida and Koichiro Yamauchi

Graduate School of Information Science and Technology, Hokkaido University,
Kita 14 Nishi 9, Kita, Sapporo, 060-0814, Japan
{knishida, yamauchi}@complex.eng.hokudai.ac.jp

Abstract. Detecting concept drift is important for dealing with real-world online learning problems. To detect concept drift in a small number of examples, methods that have an online classifier and monitor its prediction errors during the learning have been developed. We have developed such a detection method that uses a statistical test of equal proportions. Experimental results showed that our method performed well in detecting the concept drift in five synthetic datasets that contained various types of concept drift.

1 Introduction

A difficult problem in learning scenarios is that the underlying distribution of the target concept may change over time. This is generally known as "concept drift" [1]. We have developed a method to detect concept drift in an online learning scenario in which a classifier is sequentially presented with training examples. The classifier outputs a class prediction for the given input, $\boldsymbol{x}_t$, at each time step and then updates its hypothesis based on the true class label, y_t. Each example is independently drawn from the current distribution of the target concept, $\Pr_t(\boldsymbol{x}, y)$. If concept drift occurs at time t, $\Pr_t(\boldsymbol{x}, y)$ differs from $\Pr_{t-1}(\boldsymbol{x}, y)$. The task of the method is to detect changes quickly and accurately to enable the classifier to minimize cumulative prediction errors during online learning.

The detection of changes is one way to respond to concept drift. Examples of real problems where change detection is relevant include user modeling, monitoring in biomedicine and industrial processes, fault detection and diagnosis [2]. There has been much work on detecting changes in online data streams [2,3,4]; however, most of it is based on estimating the underlying distribution of examples, which requires a large number of examples.

Detection methods that monitor classification errors in an online classifier during online learning have been proposed recently [5,6,7]. These methods do not depend on the type of input attribute. Moreover, they are able to detect concept drift from a small number of examples and thus have low computational costs.

* This study was partly supported by a Grant-in-Aid for JSPS Fellows (18-4475) from the Japan Society for the Promotion of Science.

V. Corruble, M. Takeda, and E. Suzuki (Eds.): DS 2007, LNAI 4755, pp. 264–269, 2007.

We have proposed such a drift detection method that uses a statistical test of equal proportions (STEPD) to detect various types of concept drift quickly and accurately. We demonstrated experimentally the performance of the method using five synthetic datasets that contain concept drift.

2 Related Drift Detection Methods

Gama et al. proposed a drift detection method with an online classifier (DDM) [5]. For each time, t, the error rate is the probability of misclassifying, p_t, with standard deviation, $s_t = \sqrt{p_t(1-p_t)/t}$. It is assumed that p_t decreases as time advances if the target concept is stationary, and any significant increase of p_t suggests that the concept is changing. If the concept is unchanged, then the $1-\alpha$ confidence interval for p_t with $n>30$ examples is approximately $p_t \pm z_{\alpha/2} s_t$, where $z_{\alpha/2}$ denotes the $(1-\alpha/2)$th percentile of the standard normal distribution. DDM stores the values of p_t and s_t when $p_t + s_t$ reaches its minimum value (obtaining $p_{\min}$ and $s_{\min}$) and stores examples in short-term memory while $p_t + s_t \geq p_{\min} + 2s_{\min}$ is satisfied. DDM then rebuilds the classifier from the stored examples and resets all variables if $p_t + s_t \geq p_{\min} + 3s_{\min}$. DDM performs well for sudden changes; however, it has difficulties detecting gradual changes.

To improve the detection of gradual changes, Baena-García et al. developed the early drift detection method (EDDM) [6]. Their key idea is to consider the time interval (distance) between two occurrences of classification errors. They assume that any significant decrease in the distance suggests that the concept is changing. Thus, EDDM calculates the average distance between two errors, p'_t, and its standard deviation, s'_t, and stores these values when $p'_t + 2s'_t$ reaches its maximum value (obtaining $p'_{\max}$ and $s'_{\max}$). EDDM stores examples in short-term memory while $v_t\,(=(p'_t + 2s'_t)/(p'_{\max} + 2s'_{\max})) < \alpha$ is satisfied. It then rebuilds the classifier from the stored examples and resets all variables if $v_t < \beta$. Note that it starts detecting drift after 30 errors have occurred. EDDM performs well for gradual changes; however, it is not good at detecting drift in noisy examples.

We previously developed a drift detection method in a multiple classifier system [7]. We have now simplified it. This simplified method (ACED) uses only an online classifier. ACED observes the predictive accuracy of the online classifier for recent W examples, q_t, and calculates the $1-\alpha_d$ confidence interval for q_t at every time t. Our key idea is that q_t will not fall below the lower endpoint of the interval at time $t-W$, q^l_{t-W}, if the target concept is stationary. Thus, it initializes the classifier if $q_t < q^l_{t-W}$. Note that it starts detecting drift after receiving $2W$ examples. ACED is able to detect concept drift quickly when W is small; however, such small windows often cause misdetection.

3 STEPD: Detection Method Using Statistical Testing

STEPD has been developed to achieve quick and accurate detection. The basic principle is to consider two accuracies: the recent one and the overall one. We assume two things: the accuracy of a classifier for recent W examples will be equal

to the overall accuracy from the beginning of the learning if the target concept is stationary; and a significant decrease of recent accuracy suggests that the concept is changing. The test is performed by calculating the following statistic,

$$T(r_o, r_r, n_o, n_r) = \frac{|r_o/n_o - r_r/n_r| - 0.5(1/n_o + 1/n_r)}{\sqrt{\hat{p}(1-\hat{p})(1/n_o + 1/n_r)}}, \quad (1)$$

and comparing its value to the percentile of the standard normal distribution to obtain the observed significance level (P-value)[1]. r_o is the number of correct classifications among the overall n_o examples except for recent W examples, r_r is the number of correct classifications among the $W\,(=n_r)$ examples, and $\hat{p} = (r_o+r_r)/(n_o+n_r)$. If the P-value, P, is less than a significance level, then the null hypothesis $(r_o/n_o = r_r/n_r)$ is rejected and the alternative hypothesis $(r_o/n_o > r_r/n_r)$ is accepted, namely concept drift has been detected. STEPD uses two significance levels: α_w and α_d. It stores examples in short-term memory while $P < \alpha_w$ is satisfied. It then rebuilds the classifier from the stored examples and resets all variables if $P < \alpha_d$. Note that it starts detecting drift after satisfying $n_o + n_r \geq 2W$ and the stored examples are removed if $P \geq \alpha_w$.

4 Experiment and Results

We used five synthetic datasets based on sets used in other papers concerning concept drift [5,6,8]. All the datasets have two classes. Each concept has 1000 examples. The number of training examples is 4000, except for STAGGER, which has 3000. The number of test examples is 100. The training and test examples were generated randomly according to the current concept.

- STAGGER (1S). sudden. The dataset has three nominal attributes: size (*small*, *medium*, *large*), color (*red*, *blue*, *green*), and shape (*circle*, *square*, *triangle*), and has three concepts: 1) [size = *small* and color = *red*], 2) [color = *green* or shape = *circle*], and 3) [size = *medium* or *large*].
- GAUSS (2G). sudden, noisy. The examples are labeled according to two different but overlapped Gaussian, $N([0,0],1])$ and $N([2,0],4)$. The overlapping can be considered as noise. After each change, the classification is reversed.
- MIXED2 (3M). sudden, noisy. The dataset has two boolean attributes (v, w) and two continuous attributes (x, y) from $[0, 1]$. The examples are classified as positive if at least two of the three following conditions are satisfied: v, w, $y < 0.5 + 0.3\sin(3\pi x)$. After each change, the classification is reversed. Noise is introduced by switching the labels of 10% of the examples.
- CIRCLES (4C). gradual. The examples are labeled according to the condition: if an example is inside the circle, then its label is positive. The change is achieved by displacing the center of the circle $((0.2, 0.5)\quad (0.4, 0.5)\quad (0.6, 0.5)\quad (0.8, 0.5))$ and growing its radius $(0.15\quad 0.2\quad 0.25\quad 0.3)$.

[1] We should use the Fisher's exact test where sample sizes are small; however, we did not use it due to its high computational costs. The statistic in Eq. (1) is equivalent to the chi-square test with Yates's continuity correction.

Table 1. Cumulative prediction error rate with 95% confidence interval, number of drift detection (N_d), and number of required examples to detect drift correctly (N_e)

Data set	Method	IB1			Naive Bayes (NB)		
		Error Rate	N_d	N_e	Error Rate	N_d	N_e
1S	STEPD	**.0059** ± .0001	**2.000** – **.000**	**4.29**	**.0076** ± .0002	1.998 – .010	**4.89**
	DDM	.0064 ± .0001	**2.000** – .106	7.35	.0087 ± .0002	**2.000** – .370	8.94
	EDDM	.0214 ± .0000	**2.000** – **.000**	47.3	.0208 ± .0000	**2.000** – **.000**	42.0
	ACED	.0085 ± .0001	**2.000** – **.000**	11.4	.0100 ± .0002	**2.000** – .008	11.4
	Not Use	.3134 ± .0007			.3351 ± .0006		
2G	STEPD	**.1676** ± .0007	**2.966** – 1.102	**10.6**	**.1109** ± .0004	2.964 – 1.180	**7.89**
	DDM	.2039 ± .0044	2.766 – **.892**	35.2	.1347 ± .0029	**2.970** – **1.008**	26.0
	EDDM	.1898 ± .0016	2.918 – 10.56	26.4	.1250 ± .0009	2.934 – 7.682	18.0
	ACED	.1749 ± .0008	2.880 – 5.306	12.0	.1156 ± .0005	2.910 – 3.434	9.68
	Not Use	.4456 ± .0006			.4737 ± .0007		
3M	STEPD	**.2143** ± .0009	**2.968** – .932	**12.8**	**.1885** ± .0006	**2.976** – .586	**11.2**
	DDM	.2439 ± .0036	2.672 – **.748**	43.9	.2008 ± .0013	2.942 – **.364**	36.8
	EDDM	.2443 ± .0014	2.884 – 13.20	33.6	.2175 ± .0009	2.952 – 8.704	33.5
	ACED	.2262 ± .0009	2.866 – 6.690	14.1	.2043 ± .0008	2.850 – 5.604	12.8
	Not Use	.4534 ± .0007			.4864 ± .0007		
4C	STEPD	**.0286** ± .0002	**2.952** – **.190**	**26.8**	.0956 ± .0007	1.584 – **2.292**	42.5
	DDM	.0320 ± .0003	2.318 – 1.490	58.9	.1072 ± .0010	.686 – 3.450	60.9
	EDDM	.0318 ± .0002	2.618 – .462	49.5	**.0920** ± .0004	**1.588** – 7.934	50.0
	ACED	.0529 ± .0002	1.498 – .908	31.6	.1046 ± .0009	.786 – 2.952	**37.5**
	Not Use	.1365 ± .0004			.1536 ± .0005		
5H	STEPD	**.2254** ± .0012	1.406		.1182 ± .0014	2.000	
	DDM	.2361 ± .0016	.048		.1278 ± .0017	1.518	
	EDDM	.2327 ± .0013	6.834		**.1110** ± .0011	4.800	
	ACED	.2326 ± .0009	7.486		.1176 ± .0011	3.398	
	Not Use	.2465 ± .0021			.1590 ± .0028		

Notes: The prediction error rate is only calculated from the error on training data. The form of the N_d column (ex. $n-m$) means that n is the number of detection within 100 examples after each change and m is otherwise one (corresponding to the number of misdetection). We excluded misdetection in the calculation of N_e.

– HYPERP (5H). very gradual. The examples uniformly distributed in multi-dimensional space $[0,1]^{10}$ are labeled satisfying $\sum_{i=1}^{10} a_i x_i \geq a_0$ as positive. The weights of the moving hyperplane, $\{a_i\}$, which are initialized to $[-1,1]$ randomly, are updated as $a_i \leftarrow a_i + 0.001 s_i$ at each time, where $s_i \in \{-1,1\}$ is the direction of change for each weight. The threshold a_0 is calculated as $a_0 = \frac{1}{2}\sum_{i=1}^{10} a_i$ at each time. $\{s_i\}$ is reset randomly every 1000 examples.

We compared STEPD with DDM, EDDM, ACED, and classifiers that did not use any methods (Not Use). The parameters of STEPD and ACED were $W = 30$, $\alpha_d = 0.003$, and $\alpha_w = 0.05$. Those of EDDM were $\alpha = 0.95$ and $\beta = 0.90$. We used two distinct classifiers with the methods: the Weka implementations of IB1 and Naive Bayes (NB) [9]. All results were averaged over 500 trials.

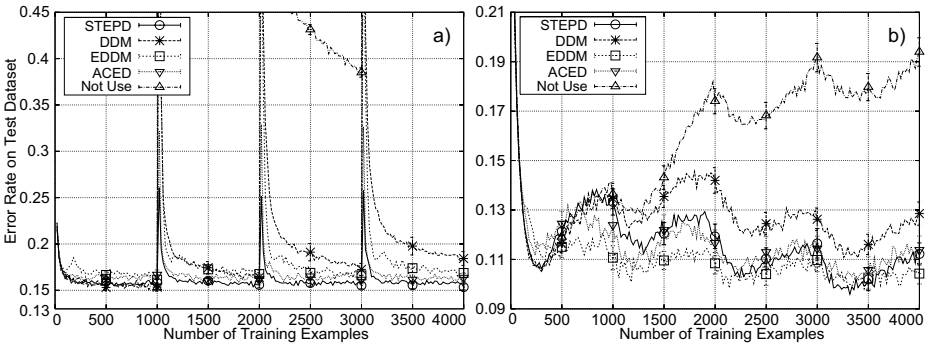

Fig. 1. Test error rate with 95% confidence intervals for a) 2G–IB1 and b) 5H–NB

Figure 1 and Table 1 show that all the detection methods improved the performance of the two classifiers in all the datasets. STEPD performed the best for sudden changes. Moreover, its performance was comparable to EDDM for gradual changes. ACED and EDDM were able to detect gradual changes well, whereas much misdetection occurred while the target concept was static because they were too sensitive to errors and noise (see N_d values for 2G and 3M). DDM detected sudden changes correctly; however, its detection speed was very slow. STEPD performed well in the presence of sudden and gradual changes and noise.

5 Conclusions

Our proposed drift detection method, STEPD, uses the statistical test of equal proportions. Experiments showed the test enables STEPD to detect various types of concept drift quickly and accurately. Future work will involve reducing misdetection and improving drift detection when changes are gradual.

References

1. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23(1), 69–101 (1996)
2. Basseville, M., Nikiforov, I.V.: Detection of Abrupt Changes: Theory and Application. Prentice-Hall, Englewood Cliffs (1993)
3. Markou, M., Singh, S.: Novelty detection: a review — part 1: Statistical approaches. Signal Processing 83(12), 2481–2497 (2003)
4. Markou, M., Singh, S.: Novelty detection: a review — part 2: Neural network based approaches. Signal Processing 83(12), 2499–2521 (2003)
5. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Proc. 17th Brazilian Symp. Artificial Intelligence, pp. 285–295 (2004)
6. Baena-García, M., del Campo-Ávila, J., Fidalgo, R., Bifet, A., Gavaldà, R., Morales-Bueno, R.: Early drift detection method. In: Proc. ECML/PKDD 2006, Work. Knowledge Discovery from Data Streams, pp. 77–86 (2006)

7. Nishida, K., Yamauchi, K., Omori, T.: ACE: Adaptive classifiers-ensemble system for concept-drifting environments. In: Proc. 6th Int. Work. Multiple Classifier Systems, pp. 176–185 (2005)
8. Wang, H., Fan, W., Yu, P.S., Han, J.: Mining concept-drifting data streams using ensemble classifiers. In: Proc. 9th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, pp. 226–235 (2003)
9. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)