Adaptive Classifiers with ICI-based Adaptive Knowledge Base Management

Cesare Alippi, Giacomo Boracchi, Manuel Roveri

Dipartimento di Elettronica e Informazione
Politecnico di Milano, Milano, Italy
{alippi, boracchi, roveri}@elet.polimi.it

Abstract. Classification systems meant to operate in non-stationary environments are requested to adapt when the process generating the observed data changes. A particularly effective form of adaptation in the abrupt perturbation case suggests to release the obsolete knowledge base of the classifier (or training set), and consider novel samples to configure the new classification model. In this direction, we propose an adaptive classifier based on a change detection test used both for detecting changes in the process and identifying the new training set (and, then, the new classifier). A key point of the proposed solution is that no assumptions are made about the distribution of the process generating the data. Experimental results show that the proposed adaptive classification system is particularly effective in situations where the process generating the data evolves through a sequence of abrupt changes.

Keywords: Adaptive Classifiers, Change Detection Tests.

1 Introduction

In the real world, data coming from industrial or environmental processes change their statistical behavior over time due to thermal drifts, aging effects, transient and permanent faults. This evolutionary nature is particularly evident in sensors subject to stress such as in X-ray detectors (due to the invasive nature of the radiation), electronic noses (due to thermal and humidity effects, as well as degradation of the active film) and monitoring system working in harsh environments (e.g., presence of water, dust, etc). Whenever a change occurs subsequent data violate that stationary hypothesis traditionally assumed in the design of the application solution, here assumed to contain a classifier. As a consequence, the classifier accuracy degrades, possibly impairing the quality of service of the application.

The need to deal with nonstationary conditions or *concept drift* [1][2] led to the development of classification systems able to adapt their knowledge base (i.e., training set) and in turn their parameters or model family to track the process evolution. In particular, [3] suggests the "instance selection" approach to trade-off accuracy and computational complexity. There, classifiers provide a classification value of a given input by relying on a subset of the knowledge base representing the current state of the process. In the same direction, FLORA and FLORA2 [4] suggest to remove a fixed 20% of the training samples (e.g., the oldest training pairs) from the knowledge base when a change is suspected (i.e., when the accuracy of the classifier decreases below a user-defined threshold). Differently, [5] suggests to adapt the knowledge base over the last samples which are assumed to contain supervised patterns.

A different and effective approach is proposed in [7] where the Just-in-Time (JIT) adaptive classifier integrates a change detection test to detect the change and an adaptive knowledge management phase removes obsolete training samples and inserts fresh ones. A soft version extends JIT classifiers to address the smooth drift case [8].

JIT classifiers consider the CI-CUSUM [6] test both to assess the stationary hypothesis and identify the training samples relevant to the classifier. Recently, a novel change detection test based on the Intersection of Confidence Interval rule (ICI) has been proposed in [9]. The test appears to be very promising as it guarantees a higher detection ability with lower detection latency and a contained computational complexity compared to the CI-CUSUM test. Moreover, the ICI test revealed to be very reliable in critical situations where only a reduced data set is available to configure the test. Whichever test we consider to detect the change, a mechanism to automatically update the test and the training set after each change detection is required.

This work provides such a mechanism by presenting a change-detection refinement procedure that adaptively identifies, once a change has been detected, the data subset representing the new process state. The novelty of the proposed approach resides in the change-detection refinement procedure which identifies the training subsequence coherent with the current state of the process. The joint use of an adaptive classifier and the change detection test allows us for improving the accuracy in stationary conditions and promptly reacting to abrupt changes in non-stationary ones.

The structure of the paper is as follows. Section II introduces the change-detection refinement procedure. The ICI-based JIT adaptive classifier dealing with both stationary and nonstationary situations is presented in Section III. Experimental results are finally given in Section IV.

2 Adaptation via Change Detection Test

Let $X : \mathbb{N} \to \mathbb{R}^d$, $d \in \mathbb{N}$ be a stochastic process generating data from two different classes of unknown *pdf*. Denote by $O_T = \{x(t), t = 1, \ldots, T\}$ the sequence of data (observations) measured up to time T, and assume that the data are independent realizations of X. Assume also that the initial T_0 observations have been generated in a stationary condition, and that the classification system uses O_{T_0} observations as training set. Since the focus is on abrupt changes, we assume that, after time T_0, the

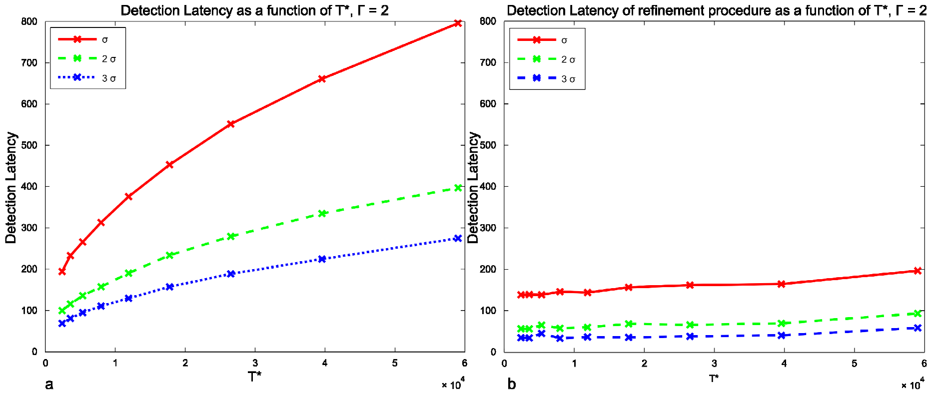

Fig. 1. Detection Latency (*DL*) as a function of process change time T^* : data are processed in subsequences of $\nu = 20$ observations, $\Gamma = 2$, and the stationary state is $X \sim \mathcal{N}(\mu, \sigma)$. Each curve represents changes in the process obtained by increasing μ of $\sigma, 2\sigma, 3\sigma$. Results have been averaged over 500 executions. **a)** The ICI change detection test (i.e., *DL* considering $\hat{T}$); **b)** the output of refined procedure (i.e., *DL* considering $T_{refined}$).

process X either does not change or evolves through a sequence of stationary states (whose change times need to be detected with a suitable test).

2.1 Detecting Changes Using the ICI rule

The change detection tests presented in [9] require a preliminary feature extraction followed by a statistical technique, the Intersection of Confidence (ICI) rule [11], [12] to assess the process stationarity.

At first we compute the sample mean and the sample variance over non overlapping subsequences of ν observations. Thanks to the Central Limit Theorem and to an ad-hoc transformation of the sample variance suggested in [13] both features z_j are Gaussian distributed

$$z_j(s) \sim \mathcal{N}\left(\mu_j(s), \sigma_j^{\,2}\right), \;\; s = 1, \ldots, T/\nu, \;\; j = 1, 2 \,, \tag{1}$$

where s indicates the subsequence index and j is the feature index. The ICI rule, combined with a polynomial smoothing operator applied to $\{z_j(s)\}_s$, is then used to identify possible changes in μ_j (i.e., the expected values for the sample mean and the transformed sample variance) and, in turn, in the stationarity of X. Experiments show that the ICI change detection test outperforms state-of-the-art solutions both in terms of reliability and computational complexity [9]. A relevant characteristic of this test is that it relies only on the tuning parameter $\Gamma > 0$, which does not depend on the change.

Algorithm 1: Change-detection refinement procedure

1. Let $\hat{T}$ be the ICI change detection test output;
2. Compute $T_1 = T_0 + (\hat{T} - T_0) / \lambda$;
3. $i = 1$; `continue = true`;
4. **`while`** (`continue == true`){
5. Apply ICI change detection test to $[0, T_0] \cup [T_i, \hat{T}]$; let $\hat{T}_i$ be the result;
6. Compute $T_{i+1} = T_i + (\hat{T} - T_i) / \lambda$;
7. **`If`** ($\min(\{\hat{T}_j\}_{j=1,\ldots,i}) < T_{i+1}$)
8.   `continue = false`; }
9. Define $T_{refined} = \min(\{\hat{T}_j\}_{j=1,\ldots,i})$; (Define $T_0 = \hat{T}$).

2.2 Change-Detection Refinement Procedure

Figure 1 a) shows the average Detection Latency (*DL*), measured as the number of observations required to detect an occurred change, over T^*, the time instant where the change occurs. It comes out that the later the change occurs, the larger is the number of observations (generated in the novel status of X) needed to detect it with the ICI detection test. Of course, this is an undesirable behavior which needs to be addressed to make the test effective in the long run. Such delays cannot be analytically compensated during an on-line data analysis, as they depend on the pdf of X before and after the change. Moreover, Figure 1 a) suggests that, once the change has been detected, the estimate of T^* can be improved by executing the ICI change detection on shorter observation sequences. This is the motivating idea of the change-detection refinement procedure, which is briefly described in the following and detailed in Algorithm 1.

Whenever the ICI change detection test reveals a process change in $\hat{T}$, the refinement procedure analyzes the previous observations to identify a more accurate estimate of the change time T^*. Operatively, the analyzed interval $[T_0, \hat{T}]$ is split in two intervals $[T_0, T_1]$ and $[T_1, \hat{T}]$ whose lengths are determined by the parameter $\lambda > 1$ (line 2), and then the ICI change detection test is run on $[0, T_0] \cup [T_1, \hat{T}]$ (line 5) providing (a possible) detection $\hat{T}_1$. This is considered a more accurate estimate of T^*, as the test operates on a shorter sequence w.r.t. the former detection. The procedure is then iterated by further splitting $[T_1, \hat{T}]$ (line 6), until the earliest detection is reached by the leftmost interval bound (line 7). An illustrative example of the change-detection refinement procedure is shown in Figure 2.

Note that this procedure provides the estimate $T_{refined}$ of T^* which is expected to be less affected by the systematic delays shown in Figure 1 a). It comes out that the

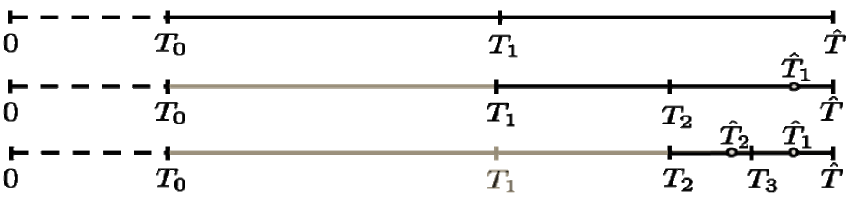

Fig. 2. Change-detection refinement procedure: an example with $\lambda = 2$. Initially (first line) a change is detected in $\hat{T}$, and the refinement starts by computing T_1. The test is thus executed on $[0,T_0] \cup [T_1,\hat{T}]$, resulting in a detection at $\hat{T}_1$ (second line). This procedure is iterated computing T_2 and running the ICI change detection test on $[0,T_0] \cup [T_2,\hat{T}]$. The procedure is terminated when $T_3 > \hat{T}_2 (= \min\{\hat{T}_j\})$. The output is $T_{refined} = \hat{T}_2$, and $[T_2,\hat{T}]$ is assumed to be generated by X in the novel (stationary) state.

observation interval $[T_{refined}, T_0]$ can be considered as being generated by X in the new stationary state. Figure 1 b) shows that the change-detection refinement procedure effectively reduces *DL* when T^* increases.

3 ICI-based Adaptive Classifier

The joint use of the ICI change detection test [9] and the change-detection refinement procedure allows us for devising a novel classification system following the philosophy of the JIT soft adaptive classifier delineated in [7]. Similarly to the JIT soft classifier, classification is performed with a *k*-NN classifier, while stationarity of *X* is monitored through the ICI change detection test.

The proposed ICI-based adaptive classifier is presented in Algorithm 2. More in detail, the sequence $Z_T = \{(x(t), y(t)), t \in I_T\}$ consists of all the supervised couples $(x(t), y(t))$ *available* and I_T contains their observation time instants. Define $I_0 = \{1,\ldots,T_0\}$ so that $Z_0 = \{(x(t), t(t)), t \in I_0\}$ is used as the initial training set for both the *k*-NN and the ICI change detection test (line 1). In particular, the initial value of *k* is estimated by means of the Leave-One-Out (LOO) technique (line 2), while the ICI change detection test is configured on the initial training set O_{T_0} (line 3).

After the initial configuration phase, the algorithm works on-line by classifying upcoming samples and by introducing, whenever available, new supervised data $(x(t), y(t))$ into the knowledge base of the classifier. In this case (line 6), the algorithm stores the time instant *t* when the sample has been received (line 7), it includes the pair $(x(t), y(t))$ in the knowledge base of the classifier (line 8) and updates the parameter *k* according to Equation (3) of [7] (line 9). In stationary conditions, the classification accuracy can be always increased by introducing additional supervised samples during the operational life [10]. When $x(t)$ carries no additional information, I_t and Z_t are not updated (lines 11-12) and $x(t)$ is classified (line 19).

Algorithm 2: ICI-based adaptive classifier (x)

1. $I_0 = \{1,..,T_0\}$, $Z_0 = \{(x(t), y(t)), t \in I_0\}$;
2. estimate k by means of LOO on Z_0 ;
3. configure the ICI change detection test using O_{T_0} ;
4. $t = T_0 + 1$;
5. **while** (1) $\{$
6.   **if** (new knowledge on $x(t)$ is available) {
7. $I_t = I_{t-1} \cup \{t\}$;
8. $Z_t = Z_{t-1} \cup \{(x(t), y(t))\}$;
9.   update k using Equation (3) of [7]. }
10.   **else** {
11. $I_t = I_{t-1}$;
12. $Z_t = Z_{t-1}$; }
13.   **if** (ICI test $(x(t))$ == " X is NOT stationary") {
14.   run the change-detection refinement procedure (Algorithm 1);
15.   configure the ICI change detection test using $[T_{refined}, T_0]$;
16.   set $I_t = \{t \in I_t \text{ , } t > T_{refined}\}$;
17.   set $Z_t = \{(x(t), y(t)), t \in I_t\}$;
18.   estimate k by means of LOO on Z_t ; }
19.   classify $x(t)$ using $k - NN(x(t), k, Z_t)$;
20. $t = t + 1;\}$

When the ICI change detection test does not identify changes in the data generating process, the current sample $x(t)$ is simply classified by the k-NN classifier (line 19) by using the current knowledge base Z_t, and the current value of k . On the contrary, when the test detects a variation in the subsequence containing $x(t)$ (line 13), the change-detection refinement procedure is executed (line 14) and produces $T_{refined}$. The change detection test is then reconfigured on the sequence $[T_{refined}, t]$ (line15), which is seen as generated by X in the novel status. This information is then exploited to remove the training samples that have been acquired before $T_{refined}$ both from I_t and from Z_t (lines 16-17). This is the main difference w.r.t. the JIT adaptive classifiers presented in [7] and [8] where the window size was either a-priori fixed by the user (as in [7]) or adapted to keep only those training samples that have been acquired in a state of the process compatible with the current one. The new value of k is then estimated by means of the LOO technique (line 18). Finally, $x(t)$ is classified by relying on the updated knowledge base (line 19).

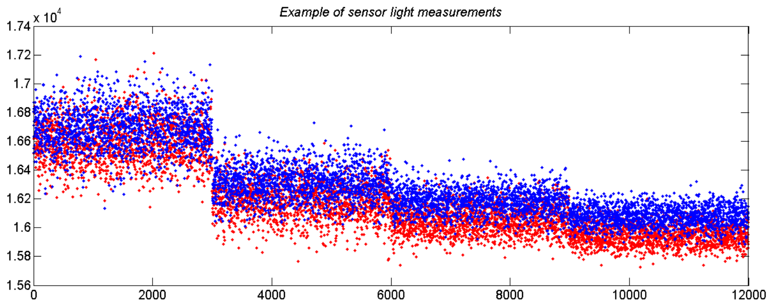

Fig. 3 An example of dataset for Application D2.

4 Experiments

The performance of the proposed adaptive classification system has been compared with those of JIT [7] and JIT soft [8] when classifying both synthetically generated data (Application D1), and measurements coming from photodiodes (Application D2).

Application D1 contains three classification datasets each of which presenting a change in stationarity: *abrupt, transient, stairs*. A dataset is composed by 200 sequences of 12000 real-valued observations drawn from two equiprobable Gaussian-distributed classes (ω_0, ω_1) that, in the initial stationary state, are distributed as $p(x \mid \omega_0) = N(0,3)$, and $p(x \mid \omega_1) = N(4,3)$. In the *abrupt* dataset, a change occurring at observation 6001 increases the mean of both classes by 15. In the *transient* dataset, the mean of both classes increases by 3 at observation 4001 and then return to the original values at observation 8001. The *stairs* dataset is characterized by a concatenation of changes at observations 3001, 6001 and 9001, each one increasing of 6 the classes' mean.

Application D2 refers to a dataset composed of 28 sequences of measurements taken from couples of photodiode sensors. Each sequence is composed of 12000 16-bit measurements (6000 per sensor). We tested the algorithms by classifying the observations according to the sensor. An example of such a sequence is shown in Figure 3.

The effectiveness of the three classifiers is measured by the classification error at time t, which corresponds to the percentage of correct classification of $x(t)$ computed over the whole dataset. Figure 4 shows these percentages averaged over a window of the 200 previous values.

We impose a minimum training set of 80 observations for the ICI-based classifier. The JIT soft has been configured with a minimum training set size of 80 observations for the classifier and 400 for the test (as required in [8]), while the JIT requires 400 observations both for the classifier and the test (as stated in [7]).

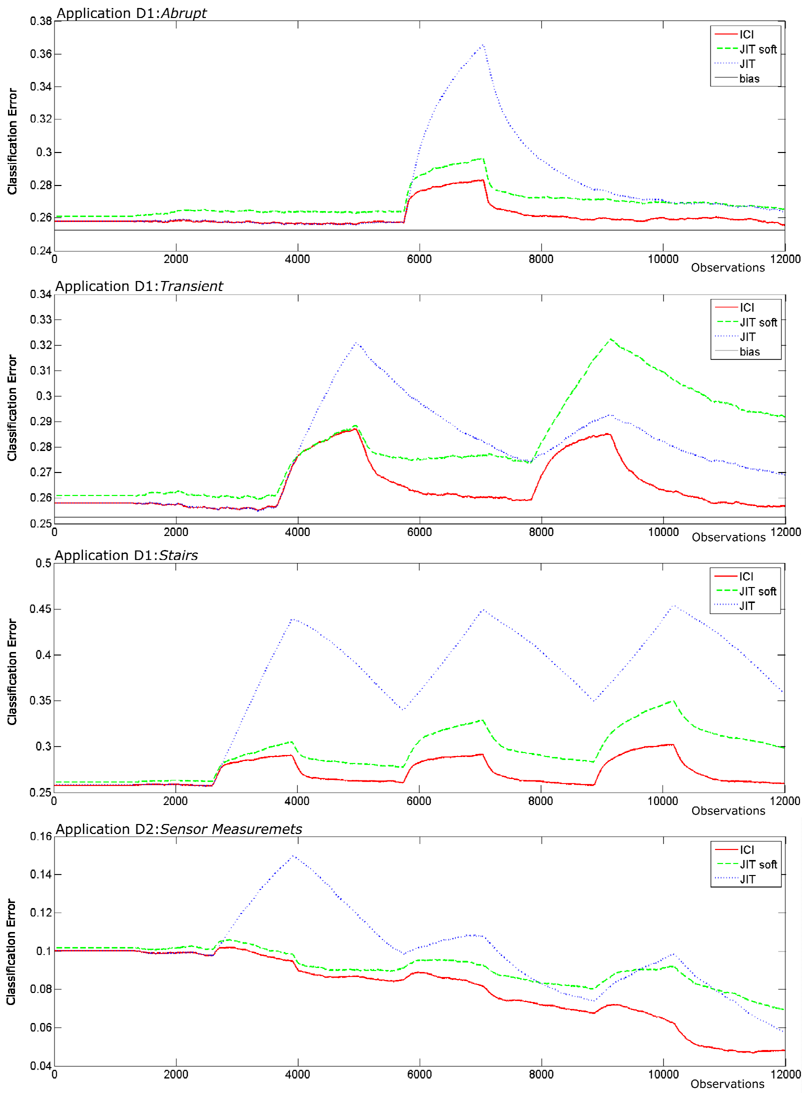

Fig. 4. Experimental results on applications D1 and D2. The classification error has been averaged over a window of 200 values

The length of the initial training set is set to $T_0 = 500$ samples; after time T_0 we provide each classifier with 1 supervised observation out of 5 to update the knowledge base. We set $\Gamma = 2$ in the ICI change detection test and $\Gamma_{refinement} = 3$ in the change-detection refinement procedure to reduce the false positives when the test is repeated several times. In the change-detection refinement procedure we also set $\lambda = 2$.

Plots of Figure 4 show a comparison among the classification errors of the three considered classifiers. In stationary conditions (i.e. before the change), the classification error typically decreases thanks to the introduction of additional supervised samples. Thus, any detection (false positive) results in an unnecessary removal of up-to-date training samples, which may significantly reduce the classification accuracy. In particular, the JIT soft shows the highest classification error due to the fact that false positives significantly reduce the training set size (this effect is less evident in JIT since, after a change is detected, the training set is composed of at least 400 samples). On the contrary, the ICI-based classifier guarantees a lower classification error since, as stated in [9], the ICI change detection test is more robust to false positives than CI-CUSUM.

In nonstationary conditions (i.e., an abrupt change occurs in the data generating process), the ICI-based classifier shows the lowest classification error thanks both to the prompter detection provided by the ICI test [9] and the change-detection refinement procedure, which identifies a timely knowledge base subset of observations representative of the new status. We emphasize that the ICI-based classifier provides an adaptive training set evolving with the process and the occurring changes, whereas in JIT classifiers the CI-CUSUM test is configured with a fixed window containing the last 400 observations. This latter, after an abrupt change, might then contain samples not coherent with the current state of the process and, hence, produce a loss in classification accuracy, as presented in Figure 4a.

When the nonstationary behavior is characterized by a sequence of abrupt changes (the *transient* and *stairs* datasets in Figures 4b and 4c), the improvement provided by the ICI-based classifier is even more evident: after the first change, the ICI-based classifier successfully adapts both the classifier and the test to the novel operating conditions and thus the test is ready to detect further changes. Conversely, after the first change, the JIT and the JIT soft cannot successfully adapt to the novel operating conditions and this affects the detection abilities on subsequent states. It is interesting to note that, in the *transient* dataset, the JIT classifier outperforms JIT soft and this is justified by the fact that the obsolete knowledge may still be present in the training set and the test configuration after the first detection.

Experiments run on photodiode sensor data (Figure 4d) shows classification errors in line with the *stairs* synthetically generated datasets.

5 Conclusions

The paper suggests an ICI adaptive classifier able to effectively react to abrupt changes in an unknown data generating process. The novel content is the definition of the change-detection refinement procedure that allows the integration of the ICI–based change detection test within the JIT framework. Such a procedure provides an effective way to identify, in nonstationary conditions, the training samples coherent with the current state of the process that can be used to configure the test and to update the knowledge base of the classifier.

Experimental results show that the proposed classification system provides higher classification accuracy than the traditional JIT and the JIT soft adaptive classifiers on both synthetically generated sequences, and light sensor measurements presenting abrupt perturbations.

References

1. Helmbold, D. P., Long P. M.: Tracking drifting concepts by minimizing disagreements. In: Machine Learning, vol. 14, no. 1, pp. 27-45, 1994.
2. Kuh, A., Petsche T.: Learning Time Varying Concepts with Applications to Pattern Recognition Problems. In: Signals, Systems and Computers, 1990. Conference Record Twenty-Fourth Asilomar Conference on,T. Petsche, Ed. vol. 2, 1990, p. 971.
3. Tsymbal. A.: The problem of concept drift: definitions and related work. Technical Report: Trinity College, Dublin, Ireland, TCD-CS-2004-15, 2004.
4. Widmer, G. Kubat, M.: Learning in the presence of concept drift and hidden contexts. In: Machine Learning, vol. 23, no. 1, pp. 69-101,1996.
5. Klinkenberg, R.: Learning drifting concepts: example selection vs. example weighting, Intelligent Data Analysis. Special Issue on Incremental Learning Systems Capable of Dealing with Concept Drift, 8 (3), 2004
6. Alippi, C., Roveri, M.: Just-in-time adaptive classifiers—Part I: Detecting nonstationarity changes. In: Neural Networks, IEEE Transactions on vol. 19, pp. 2008 .
7. Alippi, C.; Roveri, M.: Just-in-Time Adaptive Classifiers--Part II: Designing the classifier. In: Neural Networks, IEEE Transactions on, vol. 19, no. 11, pp. 2053-2064, December 2008.
8. Alippi, C., Boracchi, G., Roveri, M.: Just in time classifiers: Managing the slow drift case. In: Proc. of the IEEE 2009 International Joint Conference on Neural Networks, 2009, pp.114-120, 14-19 June 2009.
9. Alippi, C., Boracchi, G., Roveri, M.: Change Detection Tests Using the ICI rule. In: Proc. of the IEEE 2010 International Joint Conference on Neural Networks, 18-23 June 2010.
10. Fukunaga, K.: Introduction to Statistical Pattern Recognition. In: New York Academic, 1972.
11. Goldenshluger, A., Nemirovski, A.: On spatial adaptive estimation of nonparametric regression. In: Mathematical Methods of Statistics, vol 6 (1997), 135-170
12. Katkovnik, V.: A new method for varying adaptive bandwidth selection. In: Signal Processing, IEEE Transactions. on, vol. 47, no. 9, pp. 2567-2571, 1999.
13. Mudholkar G. S., Trivedi M. C.: A Gaussian Approximation to the Distribution of the Sample Variance for Nonnormal Populations. In: Journal of the American Statistical Association, Vol. 76, No. 374 (Jun., 1981), pp. 479-485