

Applying Radial Basis Function Networks and Markov Chains for on-line detection of concept drift in non-stationary environments

Ruivaldo Neto, Adrien Brilhault, Ricardo Rios

Salvador, Brazil

Abstract

The volume of data produced by computer systems has grown sharply in recent decades. A significant part of this volume is generated as uninterrupted and potentially infinite sequences known as data streams. Usually, these streams are produced by non-stationary environments, in which the data distribution can change over time. Systems applied to these environments may not be able to adapt to the new distribution, consequently deteriorating their performance. In the literature, this phenomenon is named as concept drift. Nevertheless, most concept drift detection methods are unsuited for non-stationary environments with data streams, because these algorithms often require the correct labeling of data or do not match the strict response time and resource usage requirements. In this context, this paper proposes a novel proactive on-line concept drift detection method, called RBFChain. The proposed method relies on Radial Basis Function Networks implicit clustering property and uses Markov Chains to model the transitions. To assess RBFChain as a viable concept drift detector, an analysis of sensitivity, accuracy, and noise tolerance was performed using synthetic datasets, and results were compared to the main algorithms found in the literature. Furthermore, the technique was applied to the real-world problem of eye-tracking: a critical issue in different areas of knowledge, since many behavioral experiments use eye-tracking information as a relevant analysis factor. Experimental results with synthetic datasets suggest that RBFChain is statistically better or equivalent to other detectors. Besides, the algorithm showed to apply to the eye monitoring problem, as it was able to classify fixations and saccades in real-time with accuracy comparable to state of the art.

Keywords: Concept Drift, Drift Detection, Eye tracking

1. Introduction

In recent years, the volume of data produced by computer systems has grown dramatically. This growth was favored by technological advances like the pervasiveness of mobile devices, the popularization of social networks, and the expansion of the internet of things [1]. A significant share of this data is produced in the form of uninterrupted and potentially infinite sequences. In the literature, sequences with these characteristics are known as data streams [2], and are present in various fields of application, such as financial market monitoring [3], telecom network management [4], intruder prevention [5], among others.

Most of the environments that produce data streams are non-stationary. In these environments, the data distribution can arbitrarily change over time. Systems applied to these environments may be unable to adapt to the new distribution, hence dramatically deteriorating their performance. This phenomenon is known as concept drift [6]. However, most concept drift detection methods are unfit for non-stationary environments with data streams, because these algorithms often require the correct labeling of data or do not match the strict response time and resource usage requirements.

In this context, this paper proposes a novel proactive on-line concept drift detection method, called RBFCChain. The proposed method is based on Radial Basis Function Networks implicit clustering property and employs Markov Chains to model the transitions. To validate the proposed method as a viable concept drift detector, an examination of sensitivity, accuracy, and noise tolerance was performed using synthetic datasets, and results were compared to the main algorithms found in the literature. Moreover, the algorithm was applied to the real-world problem of eye-tracking: a critical issue in different areas of knowledge, since many behavioral experiments use eye-tracking information as a relevant analysis factor.

Experimental results with synthetic datasets suggest that RBFCChain is statistically better or equivalent to other detectors as it offers greater or equal overall classification accuracy. Besides, the algorithm showed to apply to the eye monitoring problem, as it was able to classify fixations and saccades in real-time with accuracy comparable to state of the art.

The rest of the paper is organized as follows: Section 2 describes the concept drift phenomenon and the main detection techniques; Section 3 presents

the eye-tracking problem; Section 4 describes the RBFChain algorithm and its pseudo-code; Section 5 presents the setup and results of the experiment with synthetic datasets; Section 6 shows the setup and results of the eye-tracking experiment; and, finally, Section 7 provides conclusions and discusses future work.

2. Concept Drift

Most real-world problems can be regarded as non-stationary environments [6]. In these scenarios, the joint probability distribution can change over time, such as a switch in the conditional probability distribution on a classification problem, or a change of some moment (such as mean and variance) on a time series forecasting problem [7]. Systems applied to these settings may be unable to adapt to the changes, hence dramatically deteriorating their performance. This phenomenon is known as concept drift.

The concept drift phenomenon can be formally defined using Bayesian Theory [8]: consider the posterior probability of a sample x belonging to a class y , a concept drift happens when this probability changes over time, that is, $P_{t+1}(y|x) \neq P_t(y|x)$. In a supervised learning scenario, this can be interpreted as when the relationship between the input data and the target variable change over time.

According to [7, 6], concept drifts can occur in four main patterns. These patterns are demonstrated in Figure 1 and described below:

- **Abrupt:** occurs when a concept A switches abruptly to another concept B.
- **Gradual:** occurs when a concept A is being exchanged for the B concept slowly. In this case, while there is no definitive change from concept A to concept B, occurrences of B become more frequent, while fewer events of A are observed.
- **Incremental:** occurs when a concept A is being exchanged for B through intermediate concepts. These concepts differ little from its predecessor and successor, so changes are noticeable only in the long run.
- **Recurrent:** occurs when a previously active concept reappears after a certain period. However, this cannot be understood as a periodic seasonality.

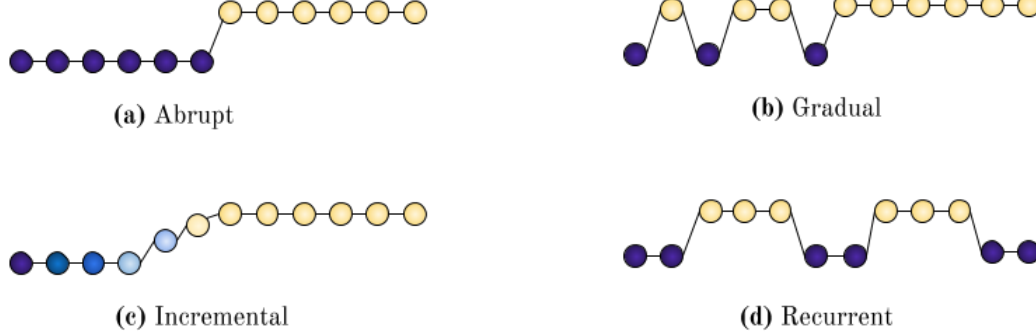


Figure 1: Concept Drift Patterns

Concept drift detection methods characterize and quantify concept drifts through the delimitation of moments or time intervals in which change happens [9]. These methods fall into two categories [10], which are described below and listed in Table 1, along with their main algorithms according to the literature.

- **Explicit Algorithms/Supervised:** Adopt a reactive approach, as they depend on the correct labeling of the data. The model performance is monitored continuously and concept drifts are detected when it starts to deteriorate, reaching a threshold.
- **Implicit Algorithms/Unsupervised:** Implement a proactive approach and are independent of data labels. Concept drifts are detected through the analysis of incoming data or indicators produced by the applied learning techniques. Although more prone to false alarms, they are an alternative to scenarios where obtaining labels is expensive, time-consuming, or unviable. Also, this approach can lead to better results, since it is possible to refit the model or adjust the data before the deterioration of the predictions.

The method proposed in this paper classifies as an *Implicit/Unsupervised* algorithm. The technique continually groups the incoming data through Radial Basis Function Networks and maintains a model of the center transitions in a Markov Chain. Concept drifts are detected when the probability of a

Table 1: Summary - Concept Drift Detection methods [11]

Category	Algorithms
Explicit/Supervised	<p>Cumulative Sum (CUSUM), PageHinkley (PH) [12], Geometric Moving Average (GMA) [13]</p> <p>Drift Detection Method (DDM) [14], Early Drift Detection Method (EDDM) [15], Exponentially Weighted Moving Average (EWMA) [16], Reactive Drift Detection Method (RDDM) [17]</p> <p>Adaptive Windowing (ADWIN) [18], SeqDrift [19], HDDMA/HDDMW [20]</p>
Implicit/Unsupervised	<p>OLINDDA [21], MINAS [22], Woo [23], DETECTNOD [24], ECSMiner [25], GC3 [26]</p> <p>CoC [27], HDDDM [28], PCA-detect [29]</p> <p>A-distance [30], CDBD [31], Margin [32]</p>

transition in the formed cluster reaches a parametric threshold. This design allows the method to detect concept drifts in real-time and independently of labels.

3. Eye-tracking

...

4. RBFChain algorithm

This section details the RBFChain implementation. However, before describing the proposed method, it is significant to present the main applied concepts of Radial Base Function Networks and Markov Chains.

100 4.1. Radial Basis Function Networks (RBFN)

101 Radial Basis Function Networks (RBFN) are used in various disciplines
 102 with a reasonable degree of success. The broad applicability is a result of
 103 their excellent ability to make function approximation, especially when the
 104 relationships among the variables of interest are nonlinear [33].

105 A radial basis function network is a type of artificial neural network
 106 (ANN), and most neural networks are known to be useful in modeling com-
 107 plex and nonlinear relationships. An RBFN has advantages in specific appli-
 108 cations in that for a given parameter set, RBFN networks do not require an
 109 iterative procedure to learn the model. Iterative learning for most ANN types
 110 is computationally expensive and vulnerable to the local minima problem.

111 The topology of an RBFN is given in Fig. 2 as a multiple input sin-
 112 gle output feedforward network. Assume that there are n input variables
 113 labeled from x_1 to x_n . The network receives input samples as vectors $x =$
 114 (x_1, x_2, \dots, x_n) of size $1 \times n$. The initial layer is only a buffer that feeds the in-
 115 put values to the intermediate layer, which is called the hidden layer. There
 116 are n_h processing elements in the hidden layer. Each processing element
 117 in the hidden layer processes the input vector and produces a single value
 118 output. This processing is performed through a basis function ϕ . Finally,
 119 the output layer weights the results of the intermediate layer by weights,
 120 aggregating them linearly to compose the final network response.

121 Among many candidates for basis functions, Gaussian radial basis func-
 122 tion (RBF), presented in Eq. 1, is used in this study. The main reason
 123 for this choice is that it can be shown that an RBFN with Gaussian RBF
 124 can sufficiently approximate any given function for a large enough number
 125 of hidden layer elements [34].

126 Probabilistic methods are built on soft decision rules, which are formalized
 127 as probabilities, e.g., the probability of a data point being a saccade given
 128 the previous observations. The probabilities and thus, the decisions are
 129 adjusted to the observations.

$$\varphi(v_i) = e^{-(\sigma r)^2} \quad (1)$$

130 In the hidden layer, each processing element has a separate vector called
 131 the center, which has the same dimensions as the input vector. For n_h
 132 hidden layer elements we have n_h center vectors as $(c_1; c_2; \dots; c_{n_h})$. Then
 133 each processing element looks at the distance between the input vector and
 134 its center and uses this distance to create its output (activation phase).

135 This work uses only the initial and intermediary layers of the presented
 136 architecture. The initial layer channels the incoming data to the middle layer,
 137 which implicitly forms clusters during the activation phase. The formed
 138 grouping has an active center that changes according to the processed value.
 139 Changes in the active center are interpreted as possible concept drifts.

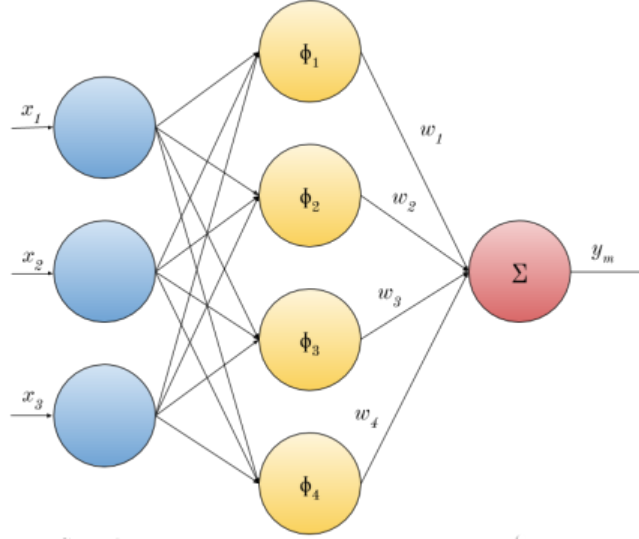


Figure 2: Topology of a RBFN

140 4.2. Markov Chains

141 A Markov chain model can be defined by the tuple $(S; A; \lambda)$. S corre-
 142 sponds to the state space, A is a matrix representing transition probabilities
 143 from one state to another, and λ is the initial probability distribution of the
 144 states in S . If there are n states in our Markov chain, then the matrix of
 145 transition probabilities A is of size $n \times n$.

146 The fundamental property of the Markov model is the dependency on the
 147 previous state. If the vector $s(t)$ denotes the probability vector for all the
 148 states at time t , then:

$$\hat{s}(t) = \hat{s}(t-1)A \quad (2)$$

149 In this proposal, Markov chains are used to model the transitions (ac-
 150 tivations) between centers in the Radial Basis Function Network. For this

151 formulation, a Markov state corresponds to one of the centers.

152 When the RBFN identifies a different center, a new state is registered in
153 the Markov Chain. Initially, all possible transitions from this center have
154 a zero value. If another center is activated, this change produces an incre-
155 ment in the probability of the correspondent transition. In paralell, all other
156 transitions probabilités are decreased proportionally to the total number of
157 possible transitions.

158 The use of a Markov Chain allows the proposed algorithm to keep an
159 online model of the transitions. The probabilities sustained in this model
160 are compared to parametric thresholds, to indicate when a warning zone is
161 triggered, or a concept drift happens.

162 4.3. *RBFChain*

163 . . .

164 5. Analyses on Synthetic Datasets with Concept Drift

165 5.1. *Experimental Setup*

166 5.2. *Results*

167 6. Detection of Saccade and Fixation

168 6.1. *Experimental Setup*

169 6.2. *Results*

170 7. Concluding Remarks

171 References

- 172 [1] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, C. Welton, Mad skills:
173 New analysis practices for big data, Proc. VLDB Endow. 2 (2009) 1481–
174 1492.
- 175 [2] C. C. Aggarwal, Data Streams: Models and Algorithms (Advances in
176 Database Systems), Springer-Verlag, Berlin, Heidelberg, 2006.
- 177 [3] L. Zhou, J. Jiang, R. Liao, T. Yang, C. Wang, Fpga based low-latency
178 market data feed handler, in: W. Xu, L. Xiao, J. Li, C. Zhang, Z. Zhu
179 (Eds.), Computer Engineering and Technology, Springer Berlin Heidel-
180 berg, Berlin, Heidelberg, 2015, pp. 69–77.

- 181 [4] M. Delattre, B. Imbert, Method for management of data stream ex-
182 changes in an autonomic telecommunications network, 2015. US Patent
183 8,949,412.
- 184 [5] P. S. Kenkre, A. Pai, L. Colaco, Real time intrusion detection and
185 prevention system, in: S. C. Satapathy, B. N. Biswal, S. K. Udgata,
186 J. Mandal (Eds.), Proceedings of the 3rd International Conference on
187 Frontiers of Intelligent Computing: Theory and Applications (FICTA)
188 2014, Springer International Publishing, Cham, 2015, pp. 405–411.
- 189 [6] K. Gama, D. Donsez, Deployment and activation of faulty components
190 at runtime for testing self-recovery mechanisms, SIGAPP Appl. Com-
191 put. Rev. 14 (2014) 44–54.
- 192 [7] A. Tsymbal, The problem of concept drift: definitions and related work,
193 Computer Science Department, Trinity College Dublin 106 (2004) 58.
- 194 [8] R. Elwell, R. Polikar, Incremental learning of concept drift in nonsta-
195 tionary environments, IEEE Transactions on Neural Networks 22 (2011)
196 1517–1531.
- 197 [9] M. Basseville, I. V. Nikiforov, Detection of Abrupt Changes: Theory and
198 Application, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
- 199 [10] I. Zliobaite, Learning under concept drift: an overview, CoRR
200 abs/1010.4784 (2010).
- 201 [11] T. S. Sethi, M. Kantardzic, On the reliable detection of concept drift
202 from streaming unlabeled data, Expert Syst. Appl. 82 (2017) 77–99.
- 203 [12] E. S. Page, Continuous Inspection Schemes, Biometrika 41 (1954) 100–
204 115.
- 205 [13] S. W. Roberts, Control chart tests based on geometric moving averages,
206 Technometrics 42 (2000) 97–101.
- 207 [14] J. Gama, P. Medas, G. Castillo, P. P. Rodrigues, Learning with drift
208 detection., in: A. L. C. Bazzan, S. Labidi (Eds.), SBIA, volume 3171 of
209 *Lecture Notes in Computer Science*, Springer, 2004, pp. 286–295.

- 210 [15] M. Baena-García, J. del Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldá,
211 R. Morales-Bueno, Early drift detection method, in: In Fourth Inter-
212 national Workshop on Knowledge Discovery from Data Streams.
- 213 [16] G. J. Ross, N. M. Adams, D. K. Tasoulis, D. J. Hand, Exponentially
214 weighted moving average charts for detecting concept drift, *Pattern*
215 *Recogn. Lett.* 33 (2012) 191–198.
- 216 [17] R. S. M. de Barros, D. R. de Lima Cabral, P. M. G. Jr., S. G. T.
217 de Carvalho Santos, RDDM: reactive drift detection method, *Expert*
218 *Syst. Appl.* 90 (2017) 344–355.
- 219 [18] A. Bifet, R. Gavald, Learning from time-changing data with adaptive
220 windowing., in: *SDM, SIAM*, 2007, pp. 443–448.
- 221 [19] R. Pears, S. Sakthithasan, Y. S. Koh, Detecting concept change in dy-
222 namic data streams - A sequential approach based on reservoir sampling,
223 *Machine Learning* 97 (2014) 259–293.
- 224 [20] I. I. F. Blanco, J. del Campo-Ávila, G. Ramos-Jiménez, R. M. Bueno,
225 A. A. O. Díaz, Y. C. Mota, Online and non-parametric drift detection
226 methods based on hoeffding’s bounds, *IEEE Trans. Knowl. Data Eng.*
227 27 (2015) 810–823.
- 228 [21] E. J. Spinosa, A. P. de Leon F. de Carvalho, J. a. Gama, Olindda: A
229 cluster-based approach for detecting novelty and concept drift in data
230 streams, in: *Proceedings of the 2007 ACM Symposium on Applied*
231 *Computing, SAC ’07*, ACM, New York, NY, USA, 2007, pp. 448–452.
- 232 [22] E. R. Faria, J. a. Gama, A. C. P. L. F. Carvalho, Novelty detection
233 algorithm for data streams multi-class problems, in: *Proceedings of the*
234 *28th Annual ACM Symposium on Applied Computing, SAC ’13*, ACM,
235 New York, NY, USA, 2013, pp. 795–800.
- 236 [23] J. W. Ryu, M. M. Kantardzic, M.-W. Kim, A. Ra Khil, An efficient
237 method of building an ensemble of classifiers in streaming data, in:
238 S. Srinivasa, V. Bhatnagar (Eds.), *Big Data Analytics*, Springer Berlin
239 Heidelberg, Berlin, Heidelberg, 2012, pp. 122–133.

- 240 [24] M. Z. Hayat, M. R. Hashemi, A dct based approach for detecting novelty
241 and concept drift in data streams, in: 2010 International Conference of
242 Soft Computing and Pattern Recognition, pp. 373–378.
- 243 [25] M. Masud, J. Gao, L. Khan, J. Han, B. M. Thuraisingham, Classification
244 and novel class detection in concept-drifting data streams under time
245 constraints, *IEEE Trans. on Knowl. and Data Eng.* 23 (2011) 859–874.
- 246 [26] T. S. Sethi, M. Kantardzic, H. Hu, A grid density based framework for
247 classifying streaming data in the presence of concept drift, *Journal of*
248 *Intelligent Information Systems* 46 (2016) 179–211.
- 249 [27] J. Lee, F. Magouls, Detection of concept drift for learning from stream
250 data, in: 2012 IEEE 14th International Conference on High Perform-
251 mance Computing and Communication 2012 IEEE 9th International
252 Conference on Embedded Software and Systems, pp. 241–245.
- 253 [28] G. Ditzler, R. Polikar, Hellinger distance based drift detection for non-
254 stationary environments, in: 2011 IEEE Symposium on Computational
255 Intelligence in Dynamic and Uncertain Environments (CIDUE), pp. 41–
256 48.
- 257 [29] L. Kuncheva, Classifier ensembles for detecting concept change in
258 streaming data: Overview and perspectives, *Proc. Eur. Conf. Artif.*
259 *Intell.* (2008) 5–10. Cited By 70.
- 260 [30] M. Dredze, T. Oates, C. Piatko, We’re not in kansas anymore: Detecting
261 domain changes in streams, pp. 585–595. Cited By 13.
- 262 [31] P. Lindstrom, B. Mac Namee, S. J. Delany, Drift detection using uncer-
263 tainty distribution divergence, *Evolving Systems* 4 (2013) 13–25.
- 264 [32] A. Dries, U. Rckert, Adaptive concept drift detection, *Statistical Analy-*
265 *sis and Data Mining: The ASA Data Science Journal* 2 (2009) 311–327.
- 266 [33] C. M. Bishop, *Pattern Recognition and Machine Learning (Information*
267 *Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.
- 268 [34] S. Theodoridis, K. Koutroumbas, *Pattern Recognition, Fourth Edition,*
269 *Academic Press, Inc., Orlando, FL, USA, 4th edition, 2008.*