

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

Ruivaldo Neto, Adrien Brilhault, Ricardo Rios

Salvador, Brazil

Abstract

Most real-world problems experience a phenomenon known as concept drift, which is a change in data distribution that can affect the system performance. However, the majority of drift detection methods are unfit for non-stationary environments with data streams. These algorithms or require the correct labeling of data (unviable in these settings) and detect drift through the monitoring of the system performance, or do not attend the severe response time and resource usage restrictions inherent to data streams. In an attempt to mitigate the aforementioned problem, this paper proposes a novel proactive method for on-line drift detection, called RBFChain. The proposed method relies on Radial Basis Networks implicit clustering property, besides using Markov Chains to model the transitions and achieve noise tolerance. To assess the proposed method 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 most established algorithms in the literature. Furthermore, the algorithm was applied to the real-world problem of eye-tracking. A relevant problem with impact in different areas of knowledge, since many behavioral experiments use eye-tracking information (fixations and saccades) as a relevant analysis factor. Performed experiments reveal that RBFChain can classify fixations and saccades in real-time, with high accuracy, noise tolerance, and using limited resources.

Keywords: Concept Drift, Drift Detection, Eye tracking

1. Introduction

In recent years, the volume of data produced by computer systems has grown dramatically. Technological advances favored this growth, such as the pervasiveness of mobile devices, the popularization of social networks and the expansion of the internet of things [1].

A significant portion of this information is produced in the form of uninterrupted and potentially infinite sequences [2]. In literature, sequences with these characteristics are called data streams and are present in various fields of application such as financial market monitoring [3], road traffic monitoring [4], telecom network management [5], real-time sentiment analysis [6] and intruder prevention and identification systems [7].

Most of the environments that produce data streams are non-stationary. That is, the joint probability distribution changes arbitrarily 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 [8]. Systems applied to these environments may be unable to adapt to the new information, hence dramatically deteriorating their performance. This phenomenon is known as concept drift [9].

Still, most concept drift detection methods adopt a reactive approach and require the correct labeling of incoming data. Generally, this approach is unfit for non-stationary environments with data streams. In these circumstances, correct labeling of data can be costly or simply unviable, and an on-line treatment of data is required since it is a continuous flow.

In an attempt to mitigate the aforementioned problem, this paper proposes a novel proactive method for on-line drift detection, called RBFChain. The proposed method relies on Radial Basis Networks implicit clustering property, besides using Markov Chains to model the transitions and achieve noise tolerance.

To assess the proposed method 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 most established algorithms in the literature.

Furthermore, the algorithm was also tested with the real-world problem of eye-tracking. This problem is relevant because many behavioral experiments use eye-tracking information (fixations and saccades) as a relevant analysis factor. Performed experiments reveal that RBFChain can classify fixations and saccades in real-time, with high accuracy, noise tolerance, and using

38 limited resources.

39 2. Concept Drift

40 Many relevant real-world problems can be considered as non-stationary
41 environments. Examples include financial market monitoring, telecom net-
42 works, intruder detection, spam filtering, among others [9].

43 In the literature, Bayesian Theory is commonly used as a background to
44 define the concept drift phenomenon formally [10]: consider the posterior
45 probability of a sample x belonging to a class y , a concept drift happens
46 when this probability changes over time, that is, $P_{t+1}(y|x) \neq P_t(y|x)$. In a
47 supervised learning scenario, this can be interpreted as when the relationship
48 between the input data and the target variable change over time.

49 According to [8, 9], concept drifts can occur in four main patterns:

- 50 • **Abrupt:** occurs when a concept A switches abruptly to another con-
51 cept B.
- 52 • **Gradual:** occurs when a concept A is being exchanged for the B con-
53 cept gradually. In this case, while there is no definitive change from
54 concept A to concept B, occurrences of B become more frequent, while
55 fewer events of A are observed.
- 56 • **Incremental:** occurs when a concept A is being exchanged for B
57 through intermediate concepts. These concepts differ little from its
58 predecessor and successor. So changes are noticeable only in the long
59 run.
- 60 • **Recurrent:** occurs when a previously active concept reappears after
61 a certain period. However, this can not be understood as a periodic
62 seasonality.

63 Figure 1 demonstrates these patterns:

64 Algorithms for detecting concept drift characterize and quantify concept
65 drifts through the delimitation of the moments or time intervals in which
66 changes occur [11]. These algorithms fall into two categories, according to
67 the need for data labeling [12]:

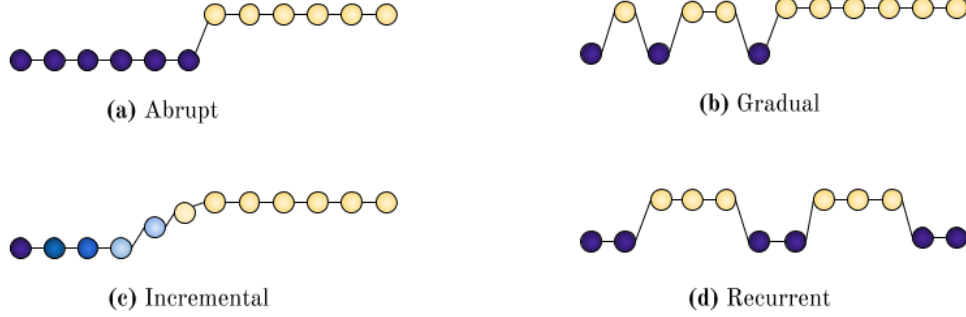


Figure 1: Concept Drift Patterns

- **Explicit Algorithms/Supervised:** These methods adopt a passive approach, as they depend on the correct labeling of the data to act. The model performance is monitored continuously, and drifts are detected when its performance starts to deteriorate, reaching a threshold.
- **Implicit Algorithms/Unsupervised:** These algorithms take a proactive approach and are independent of correct data labeling. Concept drifts are detected through the analysis of incoming data or indicators produced by the applied learning techniques. Although they are 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 algorithm proposed in this paper classifies itself as an unsupervised algorithm and adopts a proactive approach. Briefly, its operation can be described: The Radial Basis Function Networks continuously cluster all incoming data. Changes in the generated cluster (a different center is activated) reflect in a Markov Chain, which keeps an online model of the possible system transitions and its probabilities. Drifts are triggered when the transition probability reaches a parametric threshold.

To assess the proposed method as a viable concept drift detector, an analysis of sensitivity, accuracy, and noise tolerance was performed using synthetic datasets. Moreover, results were compared to the most established algorithms in the literature, demonstrating the competitiveness of the method.

91 3. Fixations and Saccades detection

92 Visual perception involves six types of eye movements [13], among which
93 fixations and saccades are the most relevant. During fixation, the eye is kept
94 relatively stable on an area of interest (AOI). In contrast, saccades are fast
95 eye movements enabling the fovea to fixate different regions of the scene [14].
96 Thus, the process of looking at a scene can be represented by a sequence of
97 fixations and saccades, the so-called visual scan path. Research on scan path
98 analysis and visual perception has benefited from the recent development of
99 eye trackers. Today's eye-tracking systems allow a precise recording of eye
100 movements at high sampling rates, thus enabling a detailed analysis of the
101 viewing behavior.

102 Despite recent advances, reliable automated clustering of eye movements
103 is still challenging, even more so in dynamic scenarios. In many applications,
104 e.g., human-computer gaze-based interaction, driving assistance systems, on-
105 line adaptation of digital content based on gaze analysis, the identification
106 of fixations and saccades has to occur in an online fashion. There is a wide
107 variety of methods for the online analysis of eye-tracking data and the recog-
108 nition of fixations and saccades. However, only a few of them are suited for
109 online applicability to dynamic scenes. Such methods have to quickly adapt
110 not only to the individual viewing behavior but also to the changes occurring
111 in the viewing scene. This small group of highly promising methods is based
112 on probabilistic formalizations, e.g., as Markov Models [15, 16], Bayesian
113 Mixture Models [17], etc.

114 Prior techniques for the automated recognition of different types of eye
115 movements from eye-tracking data fall into two main categories: (i) threshold-
116 based methods, where the distinction of fixations from saccades is based on
117 dispersion, velocity, or acceleration thresholds, and (ii) probabilistic meth-
118 ods. These groups of techniques will be briefly discussed in the following.

119 Threshold-based methods distinguish between fixations and saccades based
120 on the assumption that the distances, velocities, or accelerations occurring
121 between subsequent fixations differ from those occurring between saccades.
122 The goal then is to identify a threshold based on which saccades can be
123 reliably distinguished from fixations.

124 When distance thresholds are used, fixation clusters are usually identi-
125 fied by searching for data points that are close enough to each other (i.e.,
126 below the established threshold) within a predefined time window [18]. A
127 representative of this group, is the Dispersion Threshold Identification (I-

128 DT) algorithm [15]. Other similar approaches differ mainly in the way the
129 threshold is calculated [19, 20].

130 Other algorithms in this realm are based on the computation of Minimum
131 Spanning Trees (MST). In [15] an MST is built on the eye-tracking points
132 within a temporal window of predefined length. An edge (i.e., representing
133 the distance between two points) is classified as a saccade if its length is
134 significantly larger than the lengths of neighboring edges, which have been
135 previously classified as distances between fixations. Yet other methods em-
136 ploy smart clustering algorithms, e.g., [21, 22] but have serious limitations
137 concerning their applicability to dynamic online scenarios, since, in such sce-
138 narios, the cluster properties for fixations and saccades show high variability.

139 Methods that are based on velocity or acceleration thresholds work simi-
140 larly. A representative of this group is the Velocity-Threshold Identification
141 (I-VT) algorithm, where a point is identified as a saccade point, if the im-
142 plicit velocity along the distance from the previous data point to that point
143 exceeds a predefined threshold. Otherwise the data point is assigned to a
144 fixation cluster [15].

145 In summary, the major drawback of threshold-based methods is that they
146 rely on thresholds that have to be empirically adjusted to the individual
147 viewing behavior, the viewing area, and the specific task. Each of these pa-
148 rameters can have significant influence on the classification result [16, 15].
149 For this reason and because of the fact that the viewing behavior is strongly
150 physically and physiologically-dependent, such methods are not reliable, es-
151 pecially when real-time analysis of eye-tracking data is needed.

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

156 One of the most prominent probabilistic methods applied to the identifica-
157 tion of fixations and saccades is the Hidden Markov Model (HMM). An HMM
158 is a simple dynamic Bayesian network with variables representing values from
159 a discrete state and observation space. The state of a variable represents the
160 class of the current observation. It is only dependent on the state (i.e., class
161 of the previous observation). Because of this sequential nature, such mod-
162 els are a popular choice for the analysis of successively arising data points
163 (i.e., observations). For the detection of fixations and saccades from eye
164 data, HMMs have been used with velocity observations between successive
165 data points, thus allowing the adaptation of the model to the physiological

viewing behavior [15]. In the model of [15] (coined I-HMM), the two states used represent discretized velocity distributions over fixations and saccades. Transition probabilities between the states represent the probability of the current sample belonging to a fixation cluster or a saccade, given the previous state [18]. Due to the above probabilistic representation, no thresholds are needed. The I-HMM is reported to outperform fixed-threshold methods, such as I-VT [15]. In summary, the sequential, dynamic, and probabilistic nature of HMMs makes them an adequate choice for data arising in an online fashion and containing variability in its features.

Probabilistic mixture models, such as the Bayesian Mixture Model (BMM) presented in [17], build on the assumption that the observed data is generated from a mixture of unknown density distributions. The goal is to estimate the parameters of these distributions based on observed data points and to derive the most probable distribution that might have generated a given data point.

The algorithm presented in [17] could distinguish between fixations and saccades in an online fashion, only by considering the Euclidean distances between subsequent data points. The underlying model is based on the assumption that distances between subsequent fixation points will, in general, be shorter than distances between subsequent saccade points; that is, distances between subsequent fixation points would be generated from a specific Gaussian distribution and those between subsequent saccade points from another. This intuition was modeled by a Bayesian Online Mixture Model. The benefit of the Bayesian formalization of the mixture model is that the parameters of the two distributions are updated and learned in an online fashion as more and more data is observed. For every new data point, the prior probabilities are replaced by the latest estimates. For practical purposes, this means that for every new user the algorithm needs a relatively small number of data points to adjust to that user and learn user- or scene-dependent parameters.

In summary, probabilistic methods come with three main advantages over threshold-based ones:

1. No fixed thresholds are needed. Instead, the parameters of the model (e.g., state transition probabilities, label emission probabilities, and other settings) are learned from labeled data.
2. Both HMMs and BMMs can adapt to the individual (i.e., physiological) viewing behavior of a subject and the specific task.

203 3. Given the dynamic nature of the underlying models, the methods are
 204 naturally suited for data arising in an online fashion, such as eye-
 205 tracking data.

206 4. RBFChain algorithm

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

210 4.1. Radial Basis Function Networks (RBFN)

211 Radial Basis Function Networks (RBFN) are used in various disciplines
 212 with a reasonable degree of success. The broad applicability is a result of
 213 their excellent ability to make function approximation, especially when the
 214 relationships among the variables of interest are nonlinear [23].

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

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

231 Among many candidates for basis functions, Gaussian radial basis func-
 232 tion (RBF), presented in Eq. 1, is used in this study. The main reason
 233 for this choice is that it can be shown that an RBFN with Gaussian RBF
 234 can sufficiently approximate any given function for a large enough number
 235 of hidden layer elements [24].

236 Probabilistic methods are built on soft decision rules, which are formalized
 237 as probabilities, e.g., the probability of a data point being a saccade given

the previous observations. The probabilities and thus, the decisions are adjusted to the observations.

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

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

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

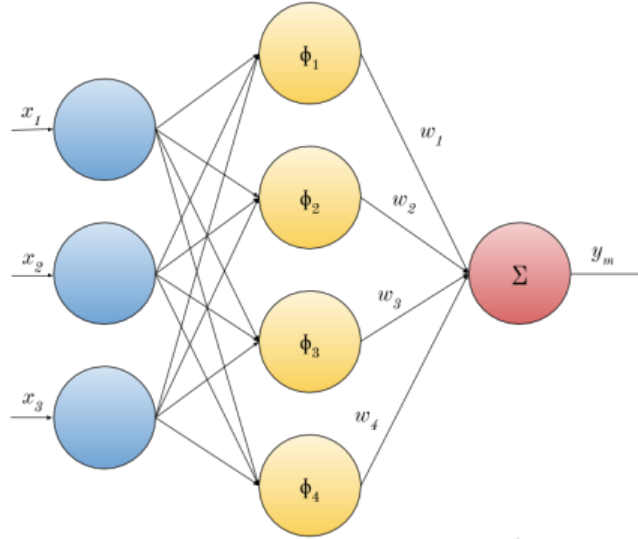


Figure 2: Topology of a RBFN

4.2. Markov Chains

A Markov chain model can be defined by the tuple $(S; A; \lambda)$. S corresponds to the state space, A is a matrix representing transition probabilities from one state to another, and λ is the initial probability distribution of the

254 states in S . If there are n states in our Markov chain, then the matrix of
 255 transition probabilities A is of size $n \times n$.

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

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

259 In this proposal, Markov chains are used to model the transitions (ac-
 260 tivations) between centers in the Radial Basis Function Network. For this
 261 formulation, a Markov state corresponds to one of the centers.

262 When the RBFN identifies a different center, a new state is registered in
 263 the Markov Chain. Initially, all possible transitions from this center have
 264 a zero value. If another center is activated, this change produces an incre-
 265 ment in the probability of the correspondent transition. In paralell, all other
 266 transitions probabilities are decreased proportionally to the total number of
 267 possible transitions.

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

272 4.3. *RBFChain*

273 ...

274 5. Analyses on Synthetic Datasets with Concept Drift

275 5.1. *Experimental Setup*

276 5.2. *Results*

277 6. Detection of Saccade and Fixation

278 6.1. *Experimental Setup*

279 6.2. *Results*

280 7. Concluding Remarks

281 References

- 282 [1] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, C. Welton, Mad skills:
 283 New analysis practices for big data, Proc. VLDB Endow. 2 (2009) 1481–
 284 1492.

- 285 [2] C. C. Aggarwal, Data Streams: Models and Algorithms (Advances in
286 Database Systems), Springer-Verlag, Berlin, Heidelberg, 2006.
- 287 [3] L. Zhou, J. Jiang, R. Liao, T. Yang, C. Wang, Fpga based low-latency
288 market data feed handler, in: W. Xu, L. Xiao, J. Li, C. Zhang, Z. Zhu
289 (Eds.), Computer Engineering and Technology, Springer Berlin Heidel-
290 berg, Berlin, Heidelberg, 2015, pp. 69–77.
- 291 [4] F. Wang, L. Hu, D. Zhou, R. Sun, J. Hu, K. Zhao, Estimating online
292 vacancies in real-time road traffic monitoring with traffic sensor data
293 stream, *Ad Hoc Netw.* 35 (2015) 3–13.
- 294 [5] M. Delattre, B. Imbert, Method for management of data stream ex-
295 changes in an autonomic telecommunications network, 2015. US Patent
296 8,949,412.
- 297 [6] J. Kranjc, J. Smailovi, V. Podpean, M. Grar, M. nidari, N. Lavra, Ac-
298 tive learning for sentiment analysis on data streams: Methodology and
299 workflow implementation in the clowdflows platform, *Information Pro-
300 cessing & Management* 51 (2015) 187 – 203.
- 301 [7] P. S. Kenkre, A. Pai, L. Colaco, Real time intrusion detection and
302 prevention system, in: S. C. Satapathy, B. N. Biswal, S. K. Udgate,
303 J. Mandal (Eds.), *Proceedings of the 3rd International Conference on
304 Frontiers of Intelligent Computing: Theory and Applications (FICTA)*
305 2014, Springer International Publishing, Cham, 2015, pp. 405–411.
- 306 [8] A. Tsymbal, The problem of concept drift: definitions and related work,
307 *Computer Science Department, Trinity College Dublin* 106 (2004) 58.
- 308 [9] K. Gama, D. Donsez, Deployment and activation of faulty components
309 at runtime for testing self-recovery mechanisms, *SIGAPP Appl. Com-
310 put. Rev.* 14 (2014) 44–54.
- 311 [10] R. Elwell, R. Polikar, Incremental learning of concept drift in nonsta-
312 tionary environments, *IEEE Transactions on Neural Networks* 22 (2011)
313 1517–1531.
- 314 [11] M. Basseville, I. V. Nikiforov, *Detection of Abrupt Changes: Theory and
315 Application*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

- 316 [12] I. Zliobaite, Learning under concept drift: an overview, CoRR
317 abs/1010.4784 (2010).
- 318 [13] R. Leigh, D. Zee, The Neurology of Eye Movements, Contemporary
319 neurology series, Oxford University Press, 2015.
- 320 [14] C. Privitera, L. Stark, Scanpath Theory, Attention, and Image Process-
321 ing Algorithms for Predicting Human Eye Fixations, pp. 296–299.
- 322 [15] D. D. Salvucci, J. H. Goldberg, Identifying fixations and saccades in
323 eye-tracking protocols, in: Proceedings of the 2000 Symposium on Eye
324 Tracking Research & Applications, ETRA '00, ACM, New York, NY,
325 USA, 2000, pp. 71–78.
- 326 [16] O. V. Komogortsev, A. Karpov, Automated classification and scoring of
327 smooth pursuit eye movements in the presence of fixations and saccades,
328 Behavior Research Methods 45 (2013) 203–215.
- 329 [17] E. Tafaj, G. Kasneci, W. Rosenstiel, M. Bogdan, Bayesian online clus-
330 tering of eye movement data, in: Proceedings of the Symposium on Eye
331 Tracking Research and Applications, ETRA '12, ACM, New York, NY,
332 USA, 2012, pp. 285–288.
- 333 [18] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, J. Halszka,
334 J. van de Weijer, Eye Tracking : A Comprehensive Guide to Methods
335 and Measures, Oxford University Press, 2011.
- 336 [19] P. Blignaut, Fixation identification: The optimum threshold for a dis-
337 persion algorithm, Attention, Perception, & Psychophysics 71 (2009)
338 881–895.
- 339 [20] F. Shic, B. Scassellati, K. Chawarska, The incomplete fixation mea-
340 sure, in: Proceedings of the 2008 Symposium on Eye Tracking Research
341 & Applications, ETRA '08, ACM, New York, NY, USA, 2008, pp.
342 111–114.
- 343 [21] A. Santella, D. DeCarlo, Robust clustering of eye movement recordings
344 for quantification of visual interest, in: Proceedings of the 2004 Sym-
345 posium on Eye Tracking Research & Applications, ETRA '04, ACM, New
346 York, NY, USA, 2004, pp. 27–34.

- 347 [22] T. Urruty, S. Lew, N. Ihadaddene, D. A. Simovici, Detecting eye fixa-
348 tions by projection clustering, *ACM Trans. Multimedia Comput. Com-*
349 *mun. Appl.* 3 (2007) 5:1–5:20.
- 350 [23] C. M. Bishop, *Pattern Recognition and Machine Learning (Information*
351 *Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.
- 352 [24] S. Theodoridis, K. Koutroumbas, *Pattern Recognition, Fourth Edition,*
353 *Academic Press, Inc., Orlando, FL, USA, 4th edition, 2008.*