

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. Generally, 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 information, 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. These algorithms usually require the correct labeling of data - infeasible in these settings - or do not match the strict response time and resource usage requirements. 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 drifts 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 most established algorithms 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. Synthetic data experimental results suggest that RBFChain is statistically better or equivalent to other detectors as it offers greater or equal overall classification accuracy. Also, the method demonstrated good performance in the eye-tracking problem, being able to identify fixations and saccades in real-time with precision comparable to state of the art.

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 share of this data is produced in the form of uninterrupted and potentially infinite sequences [2]. In the literature, sequences with these characteristics are known as data streams. This kind of stream is 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 are unsuited for non-stationary environments with data streams. These algorithms usually require the correct labeling of data - infeasible in these settings - or do not match the strict response time and resource usage requirements.

This paper proposes a novel proactive on-line concept drift detection method, called RBFChain. The proposed algorithm is based on Radial Basis Function Networks implicit clustering property and employs Markov Chains to model the drifts 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 most established algorithms in the literature. Moreover, the algorithm was also applied to the real-world problem of eye-tracking. A problem with impact in different areas of knowledge, since many behavioral experiments use eye-tracking information as a relevant analysis factor.

Experimental results suggest that RBFChain is statistically better or equivalent to other detectors as it offers greater or equal overall classification accuracy. Additionally, the method demonstrated good performance

in the eye-tracking problem, being able to identify fixations and saccades in real-time with precision 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 scenarios can be regarded as non-stationary environments [9]. In these environments, 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 [8]. 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 Bayesian Theory can be used to formally define the concept drift phenomenon [10]: 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 [8, 9], 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.

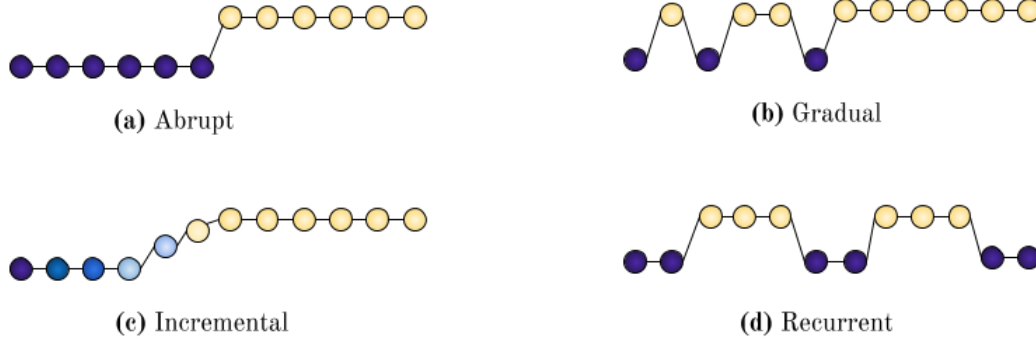


Figure 1: Concept Drift Patterns

Concept drift detection methods characterize and quantify concept drifts through the delimitation of moments or time intervals in which change happens [11]. These algorithms fall into two categories, according to the need for data labeling [12]:

- **Explicit Algorithms/Supervised:** Adopt a reactive approach, as they depend on the correct labeling of the data. The model performance is monitored continuously, and drifts are detected when its performance starts to deteriorate, passing past 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.

This paper proposes a novel unsupervised and proactive concept drift detection method. The algorithm continually groups the incoming data through Radial Basis Function Networks and maintains a model of the center transitions in a Markov Chain. Drifts are detected when the probability of a transition in the formed cluster reaches a parametric threshold.

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.

4.1. Radial Basis Function Networks (RBFN)

Radial Basis Function Networks (RBFN) are used in various disciplines with a reasonable degree of success. The broad applicability is a result of their excellent ability to make function approximation, especially when the relationships among the variables of interest are nonlinear [13].

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

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

122 Among many candidates for basis functions, Gaussian radial basis func-
 123 tion (RBF), presented in Eq. 1, is used in this study. The main reason
 124 for this choice is that it can be shown that an RBFN with Gaussian RBF
 125 can sufficiently approximate any given function for a large enough number
 126 of hidden layer elements [14].

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

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

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

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

141 4.2. Markov Chains

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

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

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

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

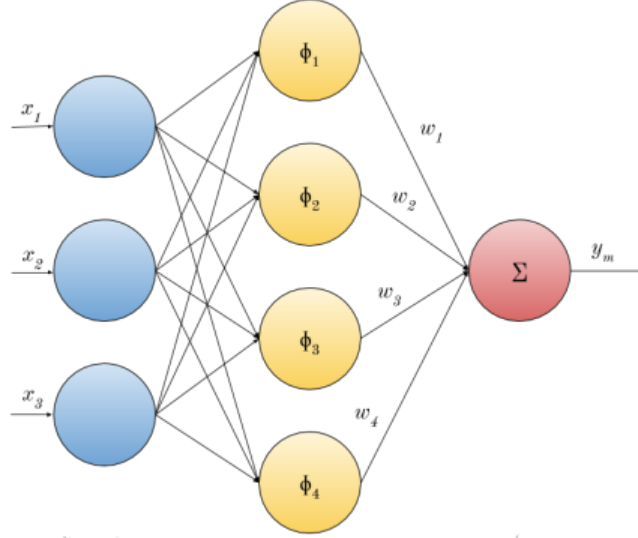


Figure 2: Topology of a RBFN

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

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

163 4.3. RBFChain

164 . . .

165 5. Analyses on Synthetic Datasets with Concept Drift

166 5.1. Experimental Setup

167 5.2. Results

168 6. Detection of Saccade and Fixation

169 6.1. Experimental Setup

170 6.2. Results

171 7. Concluding Remarks

172 References

- 173 [1] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, C. Welton, Mad skills:
174 New analysis practices for big data, Proc. VLDB Endow. 2 (2009) 1481–
175 1492.
- 176 [2] C. C. Aggarwal, Data Streams: Models and Algorithms (Advances in
177 Database Systems), Springer-Verlag, Berlin, Heidelberg, 2006.
- 178 [3] L. Zhou, J. Jiang, R. Liao, T. Yang, C. Wang, Fpga based low-latency
179 market data feed handler, in: W. Xu, L. Xiao, J. Li, C. Zhang, Z. Zhu
180 (Eds.), Computer Engineering and Technology, Springer Berlin Heidel-
181 berg, Berlin, Heidelberg, 2015, pp. 69–77.
- 182 [4] F. Wang, L. Hu, D. Zhou, R. Sun, J. Hu, K. Zhao, Estimating online
183 vacancies in real-time road traffic monitoring with traffic sensor data
184 stream, Ad Hoc Netw. 35 (2015) 3–13.
- 185 [5] M. Delattre, B. Imbert, Method for management of data stream ex-
186 changes in an autonomic telecommunications network, 2015. US Patent
187 8,949,412.
- 188 [6] J. Kranjc, J. Smailovi, V. Podpean, M. Grar, M. nidari, N. Lavra, Ac-
189 tive learning for sentiment analysis on data streams: Methodology and
190 workflow implementation in the clowdflows platform, Information Pro-
191 cessing & Management 51 (2015) 187 – 203.
- 192 [7] P. S. Kenkre, A. Pai, L. Colaco, Real time intrusion detection and
193 prevention system, in: S. C. Satapathy, B. N. Biswal, S. K. Udgata,
194 J. Mandal (Eds.), Proceedings of the 3rd International Conference on
195 Frontiers of Intelligent Computing: Theory and Applications (FICTA)
196 2014, Springer International Publishing, Cham, 2015, pp. 405–411.

- 197 [8] A. Tsymbal, The problem of concept drift: definitions and related work,
198 Computer Science Department, Trinity College Dublin 106 (2004) 58.
- 199 [9] K. Gama, D. Donsez, Deployment and activation of faulty components
200 at runtime for testing self-recovery mechanisms, SIGAPP Appl. Com-
201 put. Rev. 14 (2014) 44–54.
- 202 [10] R. Elwell, R. Polikar, Incremental learning of concept drift in nonsta-
203 tionary environments, IEEE Transactions on Neural Networks 22 (2011)
204 1517–1531.
- 205 [11] M. Basseville, I. V. Nikiforov, Detection of Abrupt Changes: Theory and
206 Application, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
- 207 [12] I. Zliobaite, Learning under concept drift: an overview, CoRR
208 abs/1010.4784 (2010).
- 209 [13] C. M. Bishop, Pattern Recognition and Machine Learning (Information
210 Science and Statistics), Springer-Verlag, Berlin, Heidelberg, 2006.
- 211 [14] S. Theodoridis, K. Koutroumbas, Pattern Recognition, Fourth Edition,
212 Academic Press, Inc., Orlando, FL, USA, 4th edition, 2008.