

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 amount of data produced by computer systems has grown sharply in recent decades, and a significant part of it 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, possibly deteriorating the system performance. In the literature, this phenomenon is named concept drift. Nevertheless, most 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 restrictions inherent to scenarios with 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 Function Networks implicit clustering property and uses Markov Chains to model the drifts transitions. 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 critical issue for 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. Also, the technique 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 portion of this data is produced in the form of uninterrupted and potentially infinite sequences [2]. In the literature, sequences with these characteristics are called data streams. These streams 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 drift detection methods are unsuitable for non-stationary environments with data streams. These methods usually require the correct labeling of data - impracticable in these contexts - or do not meet the severe response time and resource usage restrictions inherent to contexts with data streams.

This paper proposes a novel proactive method for on-line drift detection, 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 shows the configuration and results for the experiment with synthetic datasets; Section 6 presents the configuration and results for the experiment with the eye-tracking problem; and, finally, Section 7 provides conclusions and discusses future work.

2. Concept Drift

Most real-world problems 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 environments may be unable to adapt to the new information, hence dramatically deteriorating its performance. This phenomenon is called concept drift.

The Bayesian Theory can be used as a background 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:

- **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 gradually. 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 can not be understood as a periodic seasonality.

Figure 1 demonstrates these patterns:

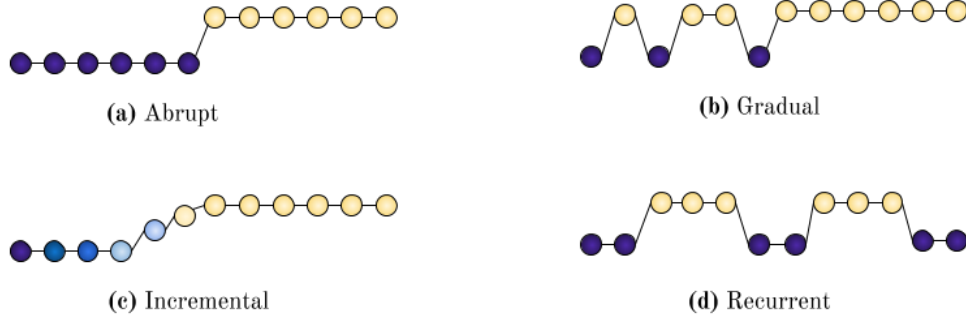


Figure 1: Concept Drift Patterns

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

- **Explicit Algorithms/Supervised:** Adopt a passive 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 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

88 false alarms, they are an alternative to scenarios where obtaining labels
89 is expensive, time-consuming or unviable. Also, this approach can lead
90 to better results, since it is possible to refit the model or adjust the
91 data before the deterioration of the predictions.

92 The algorithm proposed in this paper, RBFChain, classifies as an unsu-
93 pervised algorithm with a proactive approach.

94 **3. Eye-tracking**

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

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

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

122 Threshold-based methods distinguish between fixations and saccades based
123 on the assumption that the distances, velocities, or accelerations occurring

124 between subsequent fixations differ from those occurring between saccades.
125 The goal then is to identify a threshold based on which saccades can be
126 reliably distinguished from fixations.

127 When distance thresholds are used, fixation clusters are usually identi-
128 fied by searching for data points that are close enough to each other (i.e.,
129 below the established threshold) within a predefined time window [18]. A
130 representative of this group, is the Dispersion Threshold Identification (I-
131 DT) algorithm [15]. Other similar approaches differ mainly in the way the
132 threshold is calculated [19, 20].

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

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

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

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

159 One of the most prominent probabilistic methods applied to the identifica-
160 tion of fixations and saccades is the Hidden Markov Model (HMM). An HMM
161 is a simple dynamic Bayesian network with variables representing values from

162 a discrete state and observation space. The state of a variable represents the
 163 class of the current observation. It is only dependent on the state (i.e., class
 164 of the previous observation). Because of this sequential nature, such mod-
 165 els are a popular choice for the analysis of successively arising data points
 166 (i.e., observations). For the detection of fixations and saccades from eye
 167 data, HMMs have been used with velocity observations between successive
 168 data points, thus allowing the adaptation of the model to the physiological
 169 viewing behavior [15]. In the model of [15] (coined I-HMM), the two states
 170 used represent discretized velocity distributions over fixations and saccades.
 171 Transition probabilities between the states represent the probability of the
 172 current sample belonging to a fixation cluster or a saccade, given the previ-
 173 ous state [18]. Due to the above probabilistic representation, no thresholds
 174 are needed. The I-HMM is reported to outperform fixed-threshold methods,
 175 such as I-VT [15]. In summary, the sequential, dynamic, and probabilistic
 176 nature of HMMs makes them an adequate choice for data arising in an online
 177 fashion and containing variability in its features.

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

184 The algorithm presented in [17] could distinguish between fixations and
 185 saccades in an online fashion, only by considering the Euclidean distances
 186 between subsequent data points. The underlying model is based on the as-
 187 sumption that distances between subsequent fixation points will, in general,
 188 be shorter than distances between subsequent saccade points; that is, dis-
 189 tances between subsequent fixation points would be generated from a specific
 190 Gaussian distribution and those between subsequent saccade points from an-
 191 other. This intuition was modeled by a Bayesian Online Mixture Model. The
 192 benefit of the Bayesian formalization of the mixture model is that the param-
 193 eters of the two distributions are updated and learned in an online fashion as
 194 more and more data is observed. For every new data point, the prior prob-
 195 abilities are replaced by the latest estimates. For practical purposes, this
 196 means that for every new user the algorithm needs a relatively small num-
 197 ber of data points to adjust to that user and learn user- or scene-dependent
 198 parameters.

199 In summary, probabilistic methods come with three main advantages over

200 threshold-based ones:

- 201 1. No fixed thresholds are needed. Instead, the parameters of the model
202 (e.g., state transition probabilities, label emission probabilities, and
203 other settings) are learned from labeled data.
- 204 2. Both HMMs and BMMs can adapt to the individual (i.e., physiological)
205 viewing behavior of a subject and the specific task.
- 206 3. Given the dynamic nature of the underlying models, the methods are
207 naturally suited for data arising in an online fashion, such as eye-
208 tracking data.

209 4. RBFChain algorithm

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

213 4.1. Radial Basis Function Networks (RBFN)

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

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

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

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

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

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

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

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

253 4.2. Markov Chains

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

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

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

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

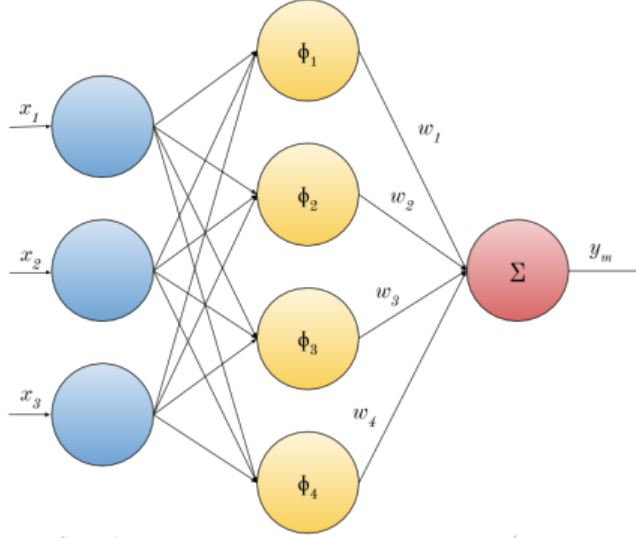


Figure 2: Topology of a RBFN

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

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

275 4.3. RBFChain

276 ...

277 5. Analyses on Synthetic Datasets with Concept Drift

278 5.1. Experimental Setup

279 5.2. Results

280 6. Detection of Saccade and Fixation

281 6.1. Experimental Setup

282 6.2. Results

283 7. Concluding Remarks

284 References

- 285 [1] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, C. Welton, Mad skills:
286 New analysis practices for big data, *Proc. VLDB Endow.* 2 (2009) 1481–
287 1492.
- 288 [2] C. C. Aggarwal, *Data Streams: Models and Algorithms (Advances in*
289 *Database Systems)*, Springer-Verlag, Berlin, Heidelberg, 2006.
- 290 [3] L. Zhou, J. Jiang, R. Liao, T. Yang, C. Wang, Fpga based low-latency
291 market data feed handler, in: W. Xu, L. Xiao, J. Li, C. Zhang, Z. Zhu
292 (Eds.), *Computer Engineering and Technology*, Springer Berlin Heidel-
293 berg, Berlin, Heidelberg, 2015, pp. 69–77.
- 294 [4] F. Wang, L. Hu, D. Zhou, R. Sun, J. Hu, K. Zhao, Estimating online
295 vacancies in real-time road traffic monitoring with traffic sensor data
296 stream, *Ad Hoc Netw.* 35 (2015) 3–13.
- 297 [5] M. Delattre, B. Imbert, Method for management of data stream ex-
298 changes in an autonomic telecommunications network, 2015. US Patent
299 8,949,412.
- 300 [6] J. Kranjc, J. Smailovi, V. Podpean, M. Grar, M. nidari, N. Lavra, Ac-
301 tive learning for sentiment analysis on data streams: Methodology and
302 workflow implementation in the clowdflows platform, *Information Pro-*
303 *cessing & Management* 51 (2015) 187 – 203.
- 304 [7] P. S. Kenkre, A. Pai, L. Colaco, Real time intrusion detection and
305 prevention system, in: S. C. Satapathy, B. N. Biswal, S. K. Udgata,
306 J. Mandal (Eds.), *Proceedings of the 3rd International Conference on*
307 *Frontiers of Intelligent Computing: Theory and Applications (FICTA)*
308 2014, Springer International Publishing, Cham, 2015, pp. 405–411.

- 309 [8] A. Tsymbal, The problem of concept drift: definitions and related work,
310 Computer Science Department, Trinity College Dublin 106 (2004) 58.
- 311 [9] K. Gama, D. Donsez, Deployment and activation of faulty components
312 at runtime for testing self-recovery mechanisms, SIGAPP Appl. Comput. Rev. 14 (2014) 44–54.
313
- 314 [10] R. Elwell, R. Polikar, Incremental learning of concept drift in nonsta-
315 tionary environments, IEEE Transactions on Neural Networks 22 (2011)
316 1517–1531.
- 317 [11] M. Basseville, I. V. Nikiforov, Detection of Abrupt Changes: Theory and
318 Application, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
- 319 [12] I. Zliobaite, Learning under concept drift: an overview, CoRR
320 abs/1010.4784 (2010).
- 321 [13] R. Leigh, D. Zee, The Neurology of Eye Movements, Contemporary
322 neurology series, Oxford University Press, 2015.
- 323 [14] C. Privitera, L. Stark, Scanpath Theory, Attention, and Image Process-
324 ing Algorithms for Predicting Human Eye Fixations, pp. 296–299.
- 325 [15] D. D. Salvucci, J. H. Goldberg, Identifying fixations and saccades in
326 eye-tracking protocols, in: Proceedings of the 2000 Symposium on Eye
327 Tracking Research & Applications, ETRA '00, ACM, New York, NY,
328 USA, 2000, pp. 71–78.
- 329 [16] O. V. Komogortsev, A. Karpov, Automated classification and scoring of
330 smooth pursuit eye movements in the presence of fixations and saccades,
331 Behavior Research Methods 45 (2013) 203–215.
- 332 [17] E. Tafaj, G. Kasneci, W. Rosenstiel, M. Bogdan, Bayesian online clus-
333 tering of eye movement data, in: Proceedings of the Symposium on Eye
334 Tracking Research and Applications, ETRA '12, ACM, New York, NY,
335 USA, 2012, pp. 285–288.
- 336 [18] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, J. Halszka,
337 J. van de Weijer, Eye Tracking : A Comprehensive Guide to Methods
338 and Measures, Oxford University Press, 2011.

- 339 [19] P. Blignaut, Fixation identification: The optimum threshold for a dis-
 340 version algorithm, *Attention, Perception, & Psychophysics* 71 (2009)
 341 881–895.
- 342 [20] F. Shic, B. Scassellati, K. Chawarska, The incomplete fixation mea-
 343 sure, in: *Proceedings of the 2008 Symposium on Eye Tracking Research*
 344 *& Applications*, ETRA '08, ACM, New York, NY, USA, 2008, pp.
 345 111–114.
- 346 [21] A. Santella, D. DeCarlo, Robust clustering of eye movement recordings
 347 for quantification of visual interest, in: *Proceedings of the 2004 Sympo-*
 348 *sium on Eye Tracking Research & Applications*, ETRA '04, ACM, New
 349 York, NY, USA, 2004, pp. 27–34.
- 350 [22] T. Urruty, S. Lew, N. Ihadaddene, D. A. Simovici, Detecting eye fixa-
 351 tions by projection clustering, *ACM Trans. Multimedia Comput. Com-*
 352 *mun. Appl.* 3 (2007) 5:1–5:20.
- 353 [23] C. M. Bishop, *Pattern Recognition and Machine Learning (Information*
 354 *Science and Statistics)*, Springer-Verlag, Berlin, Heidelberg, 2006.
- 355 [24] S. Theodoridis, K. Koutroumbas, *Pattern Recognition, Fourth Edition*,
 356 Academic Press, Inc., Orlando, FL, USA, 4th edition, 2008.