

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 unsuited for non-stationary environments with data streams. These algorithms or require the correct labeling of data - infeasible in these settings, or do not match 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 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 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

4 pervasiveness of mobile devices, the popularization of social networks and
5 the expansion of the internet of things [1].

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

12 Most of the environments that produce data streams are non-stationary.
13 That is, the joint probability distribution changes arbitrarily over time, such
14 as a switch in the conditional probability distribution on a classification prob-
15 lem, or a change of some moment (such as mean and variance) on a time series
16 forecasting problem [8]. Systems applied to these environments may be un-
17 able to adapt to the new information, hence dramatically deteriorating their
18 performance. This phenomenon is known as concept drift [9].

19 Still, most drift detection methods are improper for non-stationary envi-
20 ronments with data streams. These methods or require the correct labeling
21 of data (impracticable in these contexts) and detect drift through the mon-
22 itoring of the system performance, or do not meet the severe response time
23 and resource usage restrictions intrinsic to data streams.

24 This paper proposes a novel proactive method for on-line drift detection,
25 called RBFChain. The proposed algorithm is based on Radial Basis Net-
26 works implicit clustering property, and employs Markov Chains to model the
27 transitions and accomplish noise tolerance.

28 To validate the proposed method as a viable concept drift detector, an
29 examination of sensitivity, accuracy, and noise tolerance was performed us-
30 ing synthetic datasets, and results were compared to the most established
31 algorithms in the literature.

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

38 The rest of the paper is organized as follows: Section 2 describes the
39 concept drift phenomenon and the main detection techniques; Section 3
40 presents the fixations and saccades detection problem; Section 4 describes
41 the RBFChain algorithm and its pseudo-code; Section 5 shows the exper-

42 iment configuration for synthetic datasets and results obtained; Section 6
 43 presents the examination setup with a real-world eye-tracking problem and
 44 observed outcomes; and, finally, Section 7 provides conclusions and discusses
 45 future work.

46 2. Concept Drift

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

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

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

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

70 Figure 1 demonstrates these patterns:

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

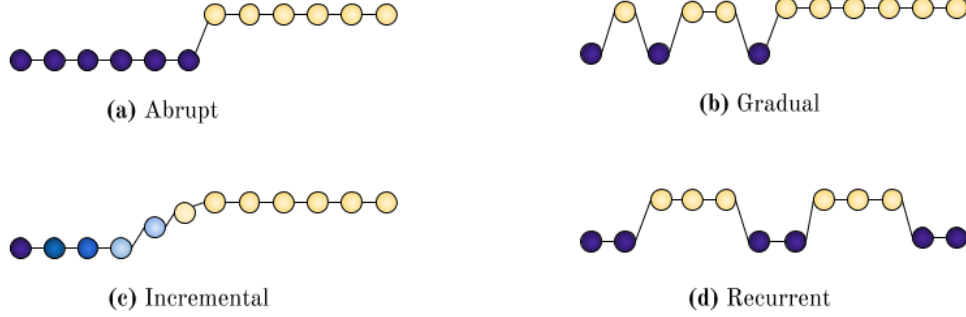


Figure 1: Concept Drift Patterns

- **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.

3. Fixations and Saccades detection

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

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

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

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

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

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

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

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

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

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

163 One of the most prominent probabilistic methods applied to the identifica-
164 tion of fixations and saccades is the Hidden Markov Model (HMM). An HMM
165 is a simple dynamic Bayesian network with variables representing values from
166 a discrete state and observation space. The state of a variable represents the
167 class of the current observation. It is only dependent on the state (i.e., class
168 of the previous observation). Because of this sequential nature, such mod-
169 els are a popular choice for the analysis of successively arising data points
170 (i.e., observations). For the detection of fixations and saccades from eye
171 data, HMMs have been used with velocity observations between successive
172 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.

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

213 4. RBFChain algorithm

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

217 4.1. Radial Basis Function Networks (RBFN)

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

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

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

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

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

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

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

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

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

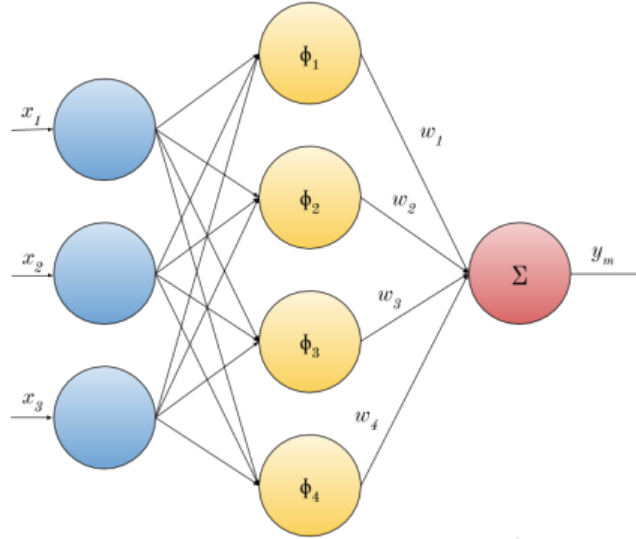


Figure 2: Topology of a RBFN

257 4.2. Markov Chains

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

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

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

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

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

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

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

279 4.3. *RBFChain*

280 ...

281 5. Analyses on Synthetic Datasets with Concept Drift

282 5.1. *Experimental Setup*

283 5.2. *Results*

284 6. Detection of Saccade and Fixation

285 6.1. *Experimental Setup*

286 6.2. *Results*

287 7. Concluding Remarks

288 References

- 289 [1] J. Cohen, B. Dolan, M. Dunlap, J. M. Hellerstein, C. Welton, Mad skills:
 290 New analysis practices for big data, Proc. VLDB Endow. 2 (2009) 1481–
 291 1492.

- 292 [2] C. C. Aggarwal, Data Streams: Models and Algorithms (Advances in
293 Database Systems), Springer-Verlag, Berlin, Heidelberg, 2006.
- 294 [3] L. Zhou, J. Jiang, R. Liao, T. Yang, C. Wang, Fpga based low-latency
295 market data feed handler, in: W. Xu, L. Xiao, J. Li, C. Zhang, Z. Zhu
296 (Eds.), Computer Engineering and Technology, Springer Berlin Heidel-
297 berg, Berlin, Heidelberg, 2015, pp. 69–77.
- 298 [4] F. Wang, L. Hu, D. Zhou, R. Sun, J. Hu, K. Zhao, Estimating online
299 vacancies in real-time road traffic monitoring with traffic sensor data
300 stream, *Ad Hoc Netw.* 35 (2015) 3–13.
- 301 [5] M. Delattre, B. Imbert, Method for management of data stream ex-
302 changes in an autonomic telecommunications network, 2015. US Patent
303 8,949,412.
- 304 [6] J. Kranjc, J. Smailovi, V. Podpean, M. Grar, M. nidari, N. Lavra, Ac-
305 tive learning for sentiment analysis on data streams: Methodology and
306 workflow implementation in the clowdflows platform, *Information Pro-
307 cessing & Management* 51 (2015) 187 – 203.
- 308 [7] P. S. Kenkre, A. Pai, L. Colaco, Real time intrusion detection and
309 prevention system, in: S. C. Satapathy, B. N. Biswal, S. K. Udgata,
310 J. Mandal (Eds.), *Proceedings of the 3rd International Conference on
311 Frontiers of Intelligent Computing: Theory and Applications (FICTA)*
312 2014, Springer International Publishing, Cham, 2015, pp. 405–411.
- 313 [8] A. Tsymbal, The problem of concept drift: definitions and related work,
314 *Computer Science Department, Trinity College Dublin* 106 (2004) 58.
- 315 [9] K. Gama, D. Donsez, Deployment and activation of faulty components
316 at runtime for testing self-recovery mechanisms, *SIGAPP Appl. Com-
317 put. Rev.* 14 (2014) 44–54.
- 318 [10] R. Elwell, R. Polikar, Incremental learning of concept drift in nonsta-
319 tionary environments, *IEEE Transactions on Neural Networks* 22 (2011)
320 1517–1531.
- 321 [11] M. Basseville, I. V. Nikiforov, *Detection of Abrupt Changes: Theory and
322 Application*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

- 323 [12] I. Zliobaite, Learning under concept drift: an overview, CoRR
324 abs/1010.4784 (2010).
- 325 [13] R. Leigh, D. Zee, The Neurology of Eye Movements, Contemporary
326 neurology series, Oxford University Press, 2015.
- 327 [14] C. Privitera, L. Stark, Scanpath Theory, Attention, and Image Process-
328 ing Algorithms for Predicting Human Eye Fixations, pp. 296–299.
- 329 [15] D. D. Salvucci, J. H. Goldberg, Identifying fixations and saccades in
330 eye-tracking protocols, in: Proceedings of the 2000 Symposium on Eye
331 Tracking Research & Applications, ETRA '00, ACM, New York, NY,
332 USA, 2000, pp. 71–78.
- 333 [16] O. V. Komogortsev, A. Karpov, Automated classification and scoring of
334 smooth pursuit eye movements in the presence of fixations and saccades,
335 Behavior Research Methods 45 (2013) 203–215.
- 336 [17] E. Tafaj, G. Kasneci, W. Rosenstiel, M. Bogdan, Bayesian online clus-
337 tering of eye movement data, in: Proceedings of the Symposium on Eye
338 Tracking Research and Applications, ETRA '12, ACM, New York, NY,
339 USA, 2012, pp. 285–288.
- 340 [18] K. Holmqvist, M. Nyström, R. Andersson, R. Dewhurst, J. Halszka,
341 J. van de Weijer, Eye Tracking : A Comprehensive Guide to Methods
342 and Measures, Oxford University Press, 2011.
- 343 [19] P. Blignaut, Fixation identification: The optimum threshold for a dis-
344 persion algorithm, Attention, Perception, & Psychophysics 71 (2009)
345 881–895.
- 346 [20] F. Shic, B. Scassellati, K. Chawarska, The incomplete fixation mea-
347 sure, in: Proceedings of the 2008 Symposium on Eye Tracking Research
348 & Applications, ETRA '08, ACM, New York, NY, USA, 2008, pp.
349 111–114.
- 350 [21] A. Santella, D. DeCarlo, Robust clustering of eye movement recordings
351 for quantification of visual interest, in: Proceedings of the 2004 Sym-
352 posium on Eye Tracking Research & Applications, ETRA '04, ACM, New
353 York, NY, USA, 2004, pp. 27–34.

- 354 [22] T. Urruty, S. Lew, N. Ihadaddene, D. A. Simovici, Detecting eye fixa-
355 tions by projection clustering, *ACM Trans. Multimedia Comput. Com-*
356 *mun. Appl.* 3 (2007) 5:1–5:20.
- 357 [23] C. M. Bishop, *Pattern Recognition and Machine Learning* (Information
358 *Science and Statistics*), Springer-Verlag, Berlin, Heidelberg, 2006.
- 359 [24] S. Theodoridis, K. Koutroumbas, *Pattern Recognition*, Fourth Edition,
360 *Academic Press, Inc.*, Orlando, FL, USA, 4th edition, 2008.