

---

# Minimax Classification under Concept Drift with Multidimensional Adaptation and Performance Guarantees

---

Verónica Álvarez<sup>1</sup> Santiago Mazuelas<sup>1,2</sup> Jose A. Lozano<sup>1,3</sup>

## Abstract

The statistical characteristics of instance-label pairs often change with time in practical scenarios of supervised classification. Conventional learning techniques adapt to such concept drift accounting for a scalar rate of change by means of a carefully chosen learning rate, forgetting factor, or window size. However, the time changes in common scenarios are multidimensional, i.e., different statistical characteristics often change in a different manner. This paper presents adaptive minimax risk classifiers (AMRCs) that account for multidimensional time changes by means of a multivariate and high-order tracking of the time-varying underlying distribution. In addition, differently from conventional techniques, AMRCs can provide computable tight performance guarantees. Experiments on multiple benchmark datasets show the classification improvement of AMRCs compared to the state-of-the-art and the reliability of the presented performance guarantees.

## 1. Introduction

The statistical characteristics describing the underlying distribution of instance-label pairs often change with time in practical scenarios of supervised classification (Gama et al., 2014; Ditzler et al., 2015). Such concept drift is common in multiple applications including electricity price prediction (Webb et al., 2018), spam mail filtering (Delany et al., 2005), and credit card fraud detection (Wang et al., 2003). For instance, in the problem of predicting electricity price

<sup>1</sup>BCAM-Basque Center for Applied Mathematics, Bilbao, Spain <sup>2</sup>IKERBASQUE-Basque Foundation for Science <sup>3</sup>Intelligent Systems Group, University of the Basque Country UPV/EHU, San Sebastián, Spain. Correspondence to: Verónica Álvarez <valvarez@bcamath.org>, Santiago Mazuelas <smazuelas@bcamath.org>, Jose A. Lozano <jlozano@bcamath.org>.

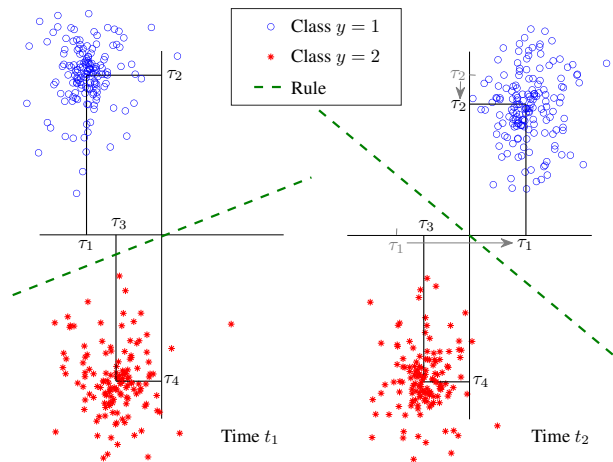


Figure 1. From time  $t_1$  to time  $t_2$ , the statistical characteristic corresponding with class 1 ( $\tau_1$  and  $\tau_2$ ) significantly change, while those corresponding with class 2 ( $\tau_3$  and  $\tau_4$ ) do not change. AMRCs account for such multidimensional time changes by tightly tracking the time-varying underlying distribution.

increases/decreases, the statistical characteristics related to electricity demand, generation, and price often change over time due to varying habits and weather. Supervised classification in those scenarios is commonly referred to as learning under concept drift (e.g., Kolter & Maloof, 2007; Webb et al., 2018), learning in a drifting (dynamic) scenario (e.g., Mohri & Medina, 2012; Shen et al., 2019), and online adaptive learning (e.g., Zhang et al., 2018; Cutkosky, 2020).

Supervised classification techniques adapt to concept drift by updating classification rules as new instance-label pairs arrive. Conventional learning techniques account for a scalar rate of change by means of a carefully chosen parameter such as a learning rate (Kivinen et al., 2004; Orabona et al., 2008), forgetting factor (Klinkenberg, 2004; Pavlidis et al., 2011), or window size (Bifet & Gavalda, 2007; Nguyen et al., 2018). Specifically, a slow/fast rate of change is tackled by using a low/high learning rate, forgetting factor, or window size. More sophisticated techniques account for a time-varying scalar rate of change by adjusting the learning rates (Shen et al., 2019), the forgetting factors (Pavlidis et al., 2011), or the window sizes

(Bifet & Gavaldà, 2007) over time. In particular, techniques based on dynamic regret minimization utilize a time-varying combination of rules obtained with different learning rates (Zhang et al., 2018; Shen et al., 2019). However, in common scenarios, the concept drift cannot be adequately addressed accounting only for a scalar rate of change. Such inadequacy is due to the fact that time changes are commonly multidimensional, i.e., different statistical characteristics of instance-label pairs often change in a different manner (see Fig. 1). For instance, in the problem of predicting electricity price increases/decreases, the statistical characteristics related to demand often change differently from those related to generation.

Conventional techniques based on statistical learning and empirical risk minimization do not provide performance guarantees under concept drift since instance-label pairs follow a time-varying underlying distribution. Techniques based on online learning and regret minimization provide performance guarantees for accumulated mistakes in terms of dynamic regret bounds (Kivinen et al., 2004; Cavallanti et al., 2007; Zhang et al., 2018; Cutkosky, 2020). Furthermore, techniques based on statistical learning for the drifting scenario provide performance guarantees for the instantaneous error probability at specific times in terms of discrepancies between consecutive distributions (Long, 1999; Mohri & Medina, 2012). However, the existing techniques offer qualitative bounds but not computable tight performance guarantees.

This paper presents adaptive minimax risk classifiers (AMRCs) that account for multidimensional time changes and provide tight performance guarantees. Specifically, the main contributions presented in the paper are as follows:

- We develop the learning methodology of AMRCs that provides multidimensional adaptation by estimating multiple statistical characteristics of the time-varying underlying distribution.
- We show that AMRCs can provide computable tight performance guarantees under concept drift in terms of instantaneous error probabilities and accumulated mistakes.
- We propose techniques to track the time-varying underlying distribution using dynamical systems that model multivariate and high-order changes in statistical characteristics.
- We propose techniques to learn AMRCs using an efficient subgradient method that utilizes warm-starts and maintains local approximations of the objective function.
- Using multiple benchmark datasets, we quantify the classification improvement of AMRCs in comparison with the state-of-the-art and numerically assess the reliability of the performance guarantees presented.

The rest of the paper is organized as follows. Section 2 briefly describes the problem formulation and minimax risk classifiers (MRCs). Section 3 presents the learning methodology of AMRCs together with their performance guarantees. We propose sequential techniques for tracking and efficient learning in Sections 4 and 5. Section 6 assesses the proposed methods using synthetic and benchmark datasets.

**Notations:** calligraphic letters represent sets; bold lowercase letters represent vectors; bold uppercase letters represent matrices;  $\mathbf{I}$  denotes the identity matrix;  $\mathbb{1}\{\cdot\}$  denotes the indicator function;  $\text{sign}(\cdot)$  denotes the vector given by the signs of the argument components;  $\|\cdot\|_1$  and  $\|\cdot\|_\infty$  denote the 1-norm and the infinity norm of its argument, respectively;  $\otimes$  and  $\odot$  denote the Kronecker and Hadamard products, respectively;  $(\cdot)_+$  and  $[\cdot]^T$  denote the positive part and the transpose of its argument, respectively;  $\preceq$  and  $\succeq$  denote vector inequalities;  $\mathbb{E}_p\{\cdot\}$  denotes the expectation of its argument with respect to distribution  $p$ ; and  $\mathbf{e}_i$  denotes the  $i$ -th vector in a standard basis.

## 2. Preliminaries

This section first recalls the basics of supervised classification in order to introduce the notation used and states the specific setting addressed. Then, we briefly describe MRC methods that minimize the worst-case error probability over an uncertainty set.

### 2.1. Problem formulation

Supervised classification uses instance-label pairs to determine classification rules that assign labels to instances. We denote by  $\mathcal{X}$  and  $\mathcal{Y}$  the sets of instances and labels, respectively; both sets are taken to be finite with  $\mathcal{Y}$  represented by  $\{1, 2, \dots, |\mathcal{Y}|\}$ . We denote by  $T(\mathcal{X}, \mathcal{Y})$  the set of all classification rules (both randomized and deterministic) and we denote by  $h(y|x)$  the probability with which rule  $h \in T(\mathcal{X}, \mathcal{Y})$  assigns label  $y \in \mathcal{Y}$  to instance  $x \in \mathcal{X}$  ( $h(y|x) \in \{0, 1\}$  for deterministic classification rules). In addition, we denote by  $\Delta(\mathcal{X} \times \mathcal{Y})$  the set of probability distributions on  $\mathcal{X} \times \mathcal{Y}$  and by  $\ell(h, p)$  the expected 0-1 loss of the classification rule  $h \in T(\mathcal{X}, \mathcal{Y})$  with respect to distribution  $p \in \Delta(\mathcal{X} \times \mathcal{Y})$ . If  $p^* \in \Delta(\mathcal{X} \times \mathcal{Y})$  is the underlying distribution of the instance-label pairs, then  $\ell(h, p^*)$  is the error probability of rule  $h$  denoted in the following as  $R(h)$ .

Instance-label pairs are usually represented as real vectors by using a feature mapping  $\Phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^m$ . The most common way to define such feature mapping is using multiple features over instances together with one-hot encodings of labels as follows (see e.g., Mohri et al., 2018)

$$\Phi(x, y) = \mathbf{e}_y \otimes \Psi(x) = [\mathbb{1}\{y = 1\}\Psi(x)^T, \mathbb{1}\{y = 2\}\Psi(x)^T, \dots, \mathbb{1}\{y = |\mathcal{Y}|\}\Psi(x)^T]^T \quad (1)$$

where the map  $\Psi : \mathcal{X} \rightarrow \mathbb{R}^d$  represents instances as real vectors. For instance, such map can be given by random Fourier features (RFF), that is

$$\Psi(x) = [\cos(\mathbf{u}_1^\top \mathbf{x}), \cos(\mathbf{u}_2^\top \mathbf{x}), \dots, \cos(\mathbf{u}_D^\top \mathbf{x}), \sin(\mathbf{u}_1^\top \mathbf{x}), \sin(\mathbf{u}_2^\top \mathbf{x}), \dots, \sin(\mathbf{u}_D^\top \mathbf{x})]^\top \quad (2)$$

for  $D$  random Gaussian vectors  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_D \sim N(0, \gamma \mathbf{I})$  with covariance given by the scaling parameter  $\gamma$  (see e.g., Lu et al., 2016; Nguyen et al., 2017; Shen et al., 2019).

In supervised classification under concept drift, samples arrive over time and the instance-label pair  $(x_t, y_t)$  at time  $t$  is a sample from an underlying distribution  $p_t \in \Delta(\mathcal{X} \times \mathcal{Y})$  that changes over time. Learning methods use each new pair to update the previous classification rule. Specifically, at each time  $t$ , these techniques predict a label  $\hat{y}_t$  corresponding with a new instance  $x_t$  using the rule  $h_t$  at time  $t$ , then they obtain updated rule  $h_{t+1}$  when the true label  $y_t$  is provided.

## 2.2. Minimax risk classifiers

MRCs learn classification rules by minimizing the worst-case error probability over distributions in an uncertainty set (Mazuelas et al., 2020; 2022a;b). Such techniques are especially suitable for supervised classification under concept drift because MRCs determine the uncertainty sets by expectation estimates and do not require to use instance-label pairs from the same underlying distribution. Other robust risk minimization (RRM) methods utilize uncertainty sets determined by metrics such as Wasserstein distances (Shafieezadeh-Abadeh et al., 2019) and f-divergences (Namkoong & Duchi, 2017). However, such techniques would require a possibly complex tracking of uncertainty sets.

MRCs learn classification rules that are solutions to

$$\min_{h \in \mathcal{T}(\mathcal{X}, \mathcal{Y})} \max_{p \in \mathcal{U}} \ell(h, p) \quad (3)$$

where  $\mathcal{U}$  is an uncertainty set of distributions. Such uncertainty set is determined by expectation estimates as

$$\mathcal{U} = \{p \in \Delta(\mathcal{X} \times \mathcal{Y}) : |\mathbb{E}_p\{\Phi(x, y)\} - \hat{\tau}| \preceq \lambda\} \quad (4)$$

where  $|\cdot|$  denotes the vector formed by the absolute value of each component in the argument,  $\hat{\tau}$  denotes the vector of expectation estimates corresponding with the feature mapping  $\Phi$ , and  $\lambda \succeq \mathbf{0}$  is a confidence vector that accounts for inaccuracies in the estimate.

As described in Mazuelas et al., 2020; 2022b using the 0-1 loss  $\ell$ , the MRC rule  $h$  solution of (3) is given by a linear combination of the feature mapping that is determined by a parameter  $\mu^*$ . Specifically, the MRC rule  $h$  assigns label

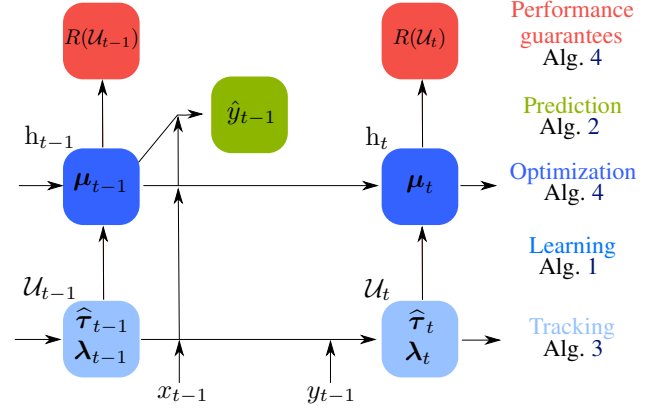


Figure 2. Diagram for the methodology of AMRCs.

$\hat{y} \in \mathcal{Y}$  to instance  $x \in \mathcal{X}$  with probability

$$h(\hat{y}|x) = \begin{cases} (\Phi(x, \hat{y})^\top \mu^* - \varphi(\mu^*))_+ / c_x & \text{if } c_x \neq 0 \\ 1/|\mathcal{Y}| & \text{if } c_x = 0 \end{cases} \quad (5)$$

with  $\varphi(\mu^*) = \max_{x \in \mathcal{X}, C \subseteq \mathcal{Y}} \left( \sum_{y \in C} \Phi(x, y)^\top \mu^* - 1 \right) / |C|$  (6)

$$c_x = \sum_{y \in \mathcal{Y}} (\Phi(x, y)^\top \mu^* - \varphi(\mu^*))_+.$$

Note that a label that maximizes the probability in (5) is given by  $\arg \max_{y \in \mathcal{Y}} h(y|x) = \arg \max_{y \in \mathcal{Y}} \Phi(x, y)^\top \mu^*$ . The deterministic classification rule  $h^d$  that assigns such label will be referred in the following as deterministic MRC.

The vector parameter  $\mu^*$  that determines the MRC rule is obtained by solving the convex optimization problem (Mazuelas et al., 2020; 2022b)

$$\min_{\mu} 1 - \hat{\tau}^\top \mu + \varphi(\mu) + \lambda^\top |\mu|. \quad (7)$$

In addition, the minimum value of (7) equals the minimax risk value of (3) that is denoted in the following as  $R(\mathcal{U})$ .

## 3. Methodology of adaptive MRCs

This section first presents the learning methodology of AMRCs. Then, we describe how AMRCs provide multidimensional adaptation and tight performance guarantees.

Figure 2 depicts the methodology of AMRCs that sequentially update MRC rules as new instance-label pairs arrive, and Algorithms 1 and 2 specify the learning and prediction stages of AMRCs. At learning, AMRCs adapt to multidimensional time changes by tightly tracking the time-varying underlying distribution (see Alg. 3); optimize the classifier parameters by using an efficient subgradient method (see Alg. 4); and obtain performance guarantees by using the minimax risk (see Alg. 4 and Section 3.2).

The tracking step updates at each time the uncertainty set  $\mathcal{U}_t$  using the uncertainty set at previous time  $\mathcal{U}_{t-1}$ . Specifically, the uncertainty set  $\mathcal{U}_t$  is determined by the mean and confidence vectors  $\hat{\tau}_t$  and  $\lambda_t$  that are updated from those at time  $t-1$  together with the instance-label pair  $(x_{t-1}, y_{t-1})$ . The optimization step updates at each time the classification rule  $h_t$  and obtains the minimax risk  $R(\mathcal{U}_t)$  using the updated uncertainty set  $\mathcal{U}_t$  together with the rule at previous time  $h_{t-1}$ . Specifically, the classification rule  $h_t$  is determined by parameter  $\mu_t$  that is obtained from the updated mean and confidence vectors  $\hat{\tau}_t$  and  $\lambda_t$  together with parameter  $\mu_{t-1}$ .

---

**Algorithm 1** Learning AMRCs
 

---

**Input:**  $(x_{t-1}, y_{t-1}), \hat{\tau}_{t-1}, \lambda_{t-1}$ , and  $\mu_{t-1}$   
**Output:**  $\hat{\tau}_t, \lambda_t, \mu_t$ , and  $R(\mathcal{U}_t)$   
 Update  $\hat{\tau}_t$  and  $\lambda_t$  using  $(x_{t-1}, y_{t-1}), \hat{\tau}_{t-1}$ , and  $\lambda_{t-1}$  (see Alg. 3)  
 Update  $\mu_t$  and obtain  $R(\mathcal{U}_t)$  solving (7) using  $\hat{\tau}_t, \lambda_t$ , and  $\mu_{t-1}$  (see Alg. 4)

---



---

**Algorithm 2** Prediction with AMRCs
 

---

**Input:**  $x_t$  and  $\mu_t$   
**Output:**  $\hat{y}_t$  for AMRC  $h_t$  or for deterministic AMRC  $h_t^d$   
 $c_x \leftarrow \sum_{y \in \mathcal{Y}} (\Phi(x_t, y)^T \mu_t - \varphi(\mu_t))_+$   
**if**  $c_x = 0$  **then**  
     **for**  $y \in \mathcal{Y}$  **do**  
          $h_t(y|x_t) \leftarrow 1/|\mathcal{Y}|$   
**else**  
     **for**  $y \in \mathcal{Y}$  **do**  
          $h_t(y|x_t) \leftarrow (\Phi(x_t, y)^T \mu_t - \varphi(\mu_t))_+ / c_x$   
 Draw  $\hat{y}_t$  from distribution  $h_t(y|x_t)$  or obtain  $\hat{y}_t$  from  $\arg \max_{y \in \mathcal{Y}} h_t(y|x_t)$

---

### 3.1. Multidimensional adaptation to time changes

The proposed learning methodology can provide multidimensional adaptation to time changes because AMRCs estimate multiple statistical characteristics of the time-varying underlying distribution.

The proposed AMRCs estimate the evolution over time of the vector formed by the expectations of the feature mapping. At each time  $t$ , such mean vector  $\tau_t = \mathbb{E}_{p_t} \{\Phi(x, y)\} \in \mathbb{R}^m$  represents the statistical characteristics of the underlying distribution  $p_t$  as measured by the feature map  $\Phi : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}^m$ . AMRCs account for situations in which different characteristics change in a different manner by estimating the evolution of each component in the mean vector.

For a feature mapping  $\Phi$  given by one-hot encodings of labels as in (1), each of the  $m = |\mathcal{Y}|d$  components of the mean vector represents one conditional expectation. For  $i = 1, 2, \dots, m$ , the  $i$ -th component  $\Phi_i(x, y)$  has associated a label  $j = 1, 2, \dots, |\mathcal{Y}|$  and an instance feature  $\Psi_r$  for

$r = 1, 2, \dots, d$ , since

$$\Phi_i(x, y) = \mathbb{1}\{y = j\} \Psi_r(x)$$

with  $i = (d-1)j + r$ . Then, the  $i$ -th component of the mean vector  $\tau_t$  corresponding to the  $j$ -th label and the  $r$ -th instance feature is given by

$$\tau_{t,i} = \mathbb{E}_{p_t} \{\Phi_i(x, y)\} = p_t(y = j) \mathbb{E}_{p_t} \{\Psi_r(x) | y = j\}$$

where  $p_t(y = j)$  denotes the probability of label  $y = j$ . The  $i$ -th component of the mean vector describes the probability of the  $j$ -th label together with the expected value of the  $r$ -th instance feature when the label takes its  $j$ -th value. In Section 4 we propose techniques for estimating each of the mean vector components accounting for multivariate and high-order time changes.

### 3.2. Performance guarantees

The proposed learning methodology can provide performance guarantees under concept drift because AMRCs minimize the worst-case error probability over distributions in a time-varying uncertainty set.

The proposed AMRCs provide performance guarantees in terms of the minimax risks obtained at learning and also with respect to the smallest minimax risks corresponding with the optimal minimax rules. Such rules correspond with uncertainty sets given by the true expectation of the feature mapping  $\Phi$ , that is

$$\mathcal{U}_t^\infty = \{p \in \Delta(\mathcal{X} \times \mathcal{Y}) : \mathbb{E}_p \{\Phi(x, y)\} = \tau_t\}. \quad (8)$$

The minimum worst-case error probability over distributions in  $\mathcal{U}_t^\infty$  is given by

$$R_t^\infty = \min_{\mu} 1 - \tau_t^T \mu + \varphi(\mu) = 1 - \tau_t^T \mu_t^\infty + \varphi(\mu_t^\infty) \quad (9)$$

corresponding with the AMRC given by parameters  $\mu_t^\infty$ . Such classification rule is referred to as optimal minimax rule because for any uncertainty set  $\mathcal{U}_t$  given by (4) that contains the underlying distribution, we have that  $\mathcal{U}_t^\infty \subseteq \mathcal{U}_t$  and hence  $R_t^\infty \leq R(\mathcal{U}_t)$ . This optimal minimax rule could only be obtained by an exact estimation of the mean vector that in turn would require an infinite amount of instance-label pairs at each time.

The following result shows performance guarantees for AMRCs both in terms of instantaneous bounds for error probabilities and bounds for accumulated mistakes.

**Theorem 1.** For  $t = 1, 2, \dots, T$ , let  $\mathcal{U}_t$  be the uncertainty set given by  $\hat{\tau} = \hat{\tau}_t$  and  $\lambda = \lambda_t$  in (4), and  $\hat{y}_t$  be a label provided by an AMRC  $h_t$  for  $\mathcal{U}_t$  given by parameter  $\mu_t$ . Then, we have that

$$R(h_t) \leq R(\mathcal{U}_t) + \alpha_t \leq R_t^\infty + \beta_t \quad (10)$$

and with probability at least  $1 - \delta$

$$\begin{aligned} \sum_{t=1}^T \mathbb{1}\{\hat{y}_t \neq y_t\} &\leq \sum_{t=1}^T R(\mathcal{U}_t) + \sum_{t=1}^T \alpha_t + \sqrt{2T \log \frac{1}{\delta}} \\ &\leq \sum_{t=1}^T R_t^\infty + \sum_{t=1}^T \beta_t + \sqrt{2T \log \frac{1}{\delta}} \end{aligned} \quad (11)$$

where  $\alpha_t$  and  $\beta_t$  can be taken as

$$\begin{aligned} \alpha_t &= \|\tau_t - \hat{\tau}_t\| - \lambda_t \|\mu_t\|_1 \\ \beta_t &= (\|\tau_t - \hat{\tau}_t\|_\infty + \|\lambda_t\|_\infty) \|\mu_t^\infty - \mu_t\|_1 \end{aligned}$$

for any  $\lambda_t \succeq \mathbf{0}$ , and as

$$\alpha_t = 0, \beta_t = 2 \|\lambda_t\|_\infty \|\mu_t^\infty\|_1$$

for  $\lambda_t \succeq |\tau_t - \hat{\tau}_t|$ .

*Proof.* See Appendix A.  $\square$

Inequalities in (10) and (11) bound the instantaneous error probability and the accumulated mistakes of AMRCs, respectively. Inequalities in (10) are obtained as a generalization of bounds in Mazuelas et al., 2020; 2022b to the addressed setting, while inequalities in (11) are obtained using the Azuma's inequality for the martingale difference  $\mathbb{1}\{\hat{y}_t \neq y_t\} - R(h_t)$ . Note that the above inequalities also ensure generalization for deterministic AMRCs because  $R(h_t^d) \leq 2R(h_t)$  since  $1 - h_t^d(y|x) \leq 2(1 - h_t(y|x))$  for any  $x \in \mathcal{X}, y \in \mathcal{Y}$ .

Existing methods based on statistical learning for the drifting scenario provide bounds for instantaneous error probabilities with respect to empirical risks (generalization bound) (Long, 1999; Mohri & Medina, 2012), while existing methods based on online learning and regret minimization provide bounds for accumulated mistakes with respect to that of comparator sequences (dynamic regret bound) (Kivinen et al., 2004; Cavallanti et al., 2007; Zhang et al., 2018; Cutkosky, 2020). Theorem 1 for AMRC methods provides bounds for instantaneous error probabilities and accumulated mistakes with respect to the smallest minimax risk  $R_t^\infty$  corresponding to the optimal minimax rule (second inequalities in (10) and (11)).

The proposed AMRCs not only provide qualitative bounds but also computable tight performance guarantees given by the minimax risk obtained at learning. However, existing methods provide bounds in terms of quantities that are not computable at learning such as discrepancies between consecutive distributions (Long, 1999; Mohri & Medina, 2012) and comparators' path-length (Zhang et al., 2018; Cutkosky, 2020). Theorem 1 for AMRC methods provides computable tight bounds for instantaneous error probabilities and accumulated mistakes (first inequalities in (10)

and (11), respectively) that are given by the minimax risk obtained by solving optimization (7) at learning. Specifically, the minimax risk  $R(\mathcal{U}_t)$  directly provides such bounds if the error in the mean vector estimate is not underestimated, i.e.,  $\lambda_t \succeq |\tau_t - \hat{\tau}_t|$ . In other cases,  $R(\mathcal{U}_t)$  still provides approximate bounds as long as the underestimation  $|\tau_t - \hat{\tau}_t| - \lambda_t$  is not substantial. In Section 6 we show that the presented bounds can provide tight performance guarantees for AMRCs in practice.

## 4. Tracking the time-varying underlying distribution

This section describes the proposed algorithms for tightly tracking the time-varying underlying distribution. Such algorithms utilize methods that are commonly used for target tracking and describe target trajectories using kinematic models (see e.g., Bar Shalom et al., 2004).

In what follows, we present techniques that estimate each component of the mean vector  $\tau_t$  at time  $t$  from instance-label pairs obtained up to time  $t - 1$ . As described in Section 3.1, for  $i = 1, 2, \dots, m$ , we denote by  $\tau_{t,i}$  the  $i$ -th component of the mean vector that corresponds with the  $j$ -th label and the  $r$ -th instance feature, that is,

$$\tau_{t,i} = \text{p}_t(y = j) \mathbb{E}_{\text{p}_t} \{\Psi_r(x) | y = j\}$$

with  $i = (d-1)j + r$  for  $j = 1, 2, \dots, |\mathcal{Y}|$  and  $r = 1, 2, \dots, d$ . Then, the estimation of each component of the mean vector is obtained from the estimation of the corresponding label probability and instance feature conditional expectation. In particular, the  $j$ -th label probability for  $j = 1, 2, \dots, |\mathcal{Y}|$  is estimated using the  $W$  latest labels as

$$\hat{\text{p}}_t(y = j) = \frac{1}{W} \sum_{i=t-W}^{t-1} \mathbb{1}\{y_i = j\}. \quad (12)$$

The above conditional expectation denoted by  $\gamma_{t,i} = \mathbb{E}_{\text{p}_t} \{\Psi_r(x) | y = j\}$  for  $i = 1, 2, \dots, m$  is recursively estimated as described in the following.

We assume that  $\gamma_{t,i}$  is  $k$  times differentiable with respect to time and denote by  $\eta_{t,i} \in \mathbb{R}^{k+1}$  the vector composed by  $\gamma_{t,i}$  and its successive derivatives up to order  $k$ . As is usually done for target tracking (Bar Shalom et al., 2004), we model the evolution of such state vector  $\eta_{t,i}$  using the partially-observed linear dynamical system

$$\begin{aligned} \eta_{t,i} &= \mathbf{H}_t \eta_{t-1,i} + \mathbf{w}_{t,i} \\ \Phi_i(x_t, y_t) &= \gamma_{t,i} + v_{t,i}, \quad \text{if } y_t = j \end{aligned} \quad (13)$$

with transition matrix  $\mathbf{H}_t = \mathbf{I} + \sum_{s=1}^k \Delta_t^s \mathbf{U}_s / s!$  where  $\mathbf{U}_s$  is the  $(k+1) \times (k+1)$  matrix with ones on the  $s$ -th upper diagonal and zeros in the rest of components,  $\Delta_t$  is the time

increment at  $t$ ,  $j = 1, 2, \dots, |\mathcal{Y}|$ , and  $i = (d-1)j + r$  for  $r = 1, 2, \dots, d$ . The variables  $\mathbf{w}_{t,i}$  and  $v_{t,i}$  represent uncorrelated noise processes with mean zero and variance  $\mathbf{Q}_{t,i}$  and  $r_{t,i}^2$ , respectively. Such variances can be estimated online using methods such as those proposed in Odelson et al., 2006; Akhlaghi et al., 2017. Dynamical systems as that given by (13) are known in target tracking as kinematic state models and can be derived using the  $k$ -th order Taylor expansion of  $\gamma_{t,i}$  (see Appendix B for a detailed derivation).

The result below allows to recursively obtain state vector estimates  $\hat{\boldsymbol{\eta}}_{t,i}$  with minimum mean squared error (MSE) together with their MSE matrices  $\hat{\boldsymbol{\Sigma}}_{t,i}$ , for  $i = 1, 2, \dots, m$ . Then, the first components of such vector and matrix provide the estimate and confidence for the conditional expectations  $\gamma_{t,i}$  for  $i = 1, 2, \dots, m$ .

**Theorem 2.** If the evolution of the state vector  $\boldsymbol{\eta}_{t,i}$  is given by the dynamical system in (13), then the linear estimator of  $\boldsymbol{\eta}_{t,i}$  based on  $(x_1, y_1), (x_2, y_2), \dots, (x_{t-1}, y_{t-1})$  that has the minimum MSE is given by the recursion

$$\hat{\boldsymbol{\eta}}_{t,i} = \mathbf{H}_t \hat{\boldsymbol{\eta}}_{t-1,i} - (\hat{\gamma}_{t-1,i} - \Phi_i(x_{t-1}, y_{t-1})) \mathbf{k}_{t,i} \quad (14)$$

$$\hat{\boldsymbol{\Sigma}}_{t,i} = \mathbf{H}_t \hat{\boldsymbol{\Sigma}}_{t-1,i} \mathbf{H}_t^\top + \mathbf{Q}_{t,i} - \mathbf{k}_{t,i} \mathbf{e}_1^\top \hat{\boldsymbol{\Sigma}}_{t-1,i} \mathbf{H}_t^\top \quad (15)$$

$$\text{where } \mathbf{k}_{t,i} = \mathbb{1}\{y_{t-1} = j\} \frac{\mathbf{H}_t \hat{\boldsymbol{\Sigma}}_{t-1,i} \mathbf{e}_1}{\mathbf{e}_1^\top \hat{\boldsymbol{\Sigma}}_{t-1,i} \mathbf{e}_1 + r_{t-1,i}^2}. \quad (16)$$

In addition,  $\hat{\boldsymbol{\Sigma}}_{t,i}$  is the MSE matrix of such estimator  $\hat{\boldsymbol{\eta}}_{t,i}$ .

*Proof.* The unbiased linear estimator with minimum MSE for a dynamical system such as (13) is given by the Kalman filter recursions (see e.g., Humpherys et al., 2012). Then, equations (14) and (15) are obtained after some algebra from the Kalman recursions for predicted state vector and predicted MSE.  $\square$

The above theorem enables to tightly track the time-varying underlying distribution. Specifically, Theorem 2 allows to recursively obtain mean vector estimates  $\hat{\boldsymbol{\tau}}_t$  as well as its confidence vectors  $\boldsymbol{\lambda}_t$  every time a new instance-label pair is received. Such vectors are obtained from the estimated label probabilities in equation (12) together with the estimated state vector and its MSE in recursions (14) and (15) as follows

$$\hat{\boldsymbol{\tau}}_{t,i} = \hat{p}_t(y = j) \mathbf{e}_1^\top \hat{\boldsymbol{\eta}}_{t,i} = \hat{p}_t(y = j) \hat{\gamma}_{t,i} \quad (17)$$

$$\begin{aligned} \boldsymbol{\lambda}_{t,i}^2 &= \hat{p}_t(y = j) (1 - \hat{p}_t(y = j)) \mathbf{e}_1^\top \hat{\boldsymbol{\Sigma}}_{t,i} \mathbf{e}_1 \\ &+ \hat{p}_t(y = j)^2 \mathbf{e}_1^\top \hat{\boldsymbol{\Sigma}}_{t,i} \mathbf{e}_1 + \hat{\gamma}_{t,i}^2 \hat{p}_t(y = j) (1 - \hat{p}_t(y = j)) \\ &= \hat{p}_t(y = j) \left( \hat{\gamma}_{t,i}^2 (1 - \hat{p}_t(y = j)) + \mathbf{e}_1^\top \hat{\boldsymbol{\Sigma}}_{t,i} \mathbf{e}_1 \right) \quad (18) \end{aligned}$$

with  $i = (d-1)j + r$  for  $j = 1, 2, \dots, |\mathcal{Y}|$ ,  $r = 1, 2, \dots, d$ .

**Remark.** The proposed techniques account for multivariate and high-order changes in statistical characteristics since the dynamical systems used model the detailed evolution of each component in the mean vector. Specifically, for  $t = 1, 2, \dots$  and  $i = 1, 2, \dots, m$ , the  $i$ -th component of the mean vector estimate is updated at time  $t$  by using the corresponding dynamical model in (13), the estimate at previous time, and the most recent instance-label pair. Such update accounts for the specific evolution at time  $t$  of the  $i$ -th component of the mean vector through the recursion in equation (14) and gain vector  $\mathbf{k}_{t,i}$  in equation (16). In particular, updates for mean vector components and times with a low gain slightly change the estimate at previous time, while those updates for components and times with a high gain increase the relevance of the most recent instance-label pair.

Algorithm 3 details the proposed procedure to track the time-varying underlying distribution. Such algorithm has computational complexity  $O(mk^3)$  and memory complexity  $O(mk^2)$  where  $m = |\mathcal{Y}|d$  is the length of the feature mapping and  $k$  is the order of the dynamical system in (13). In Section 6, we use dynamical systems with orders  $k = 0, 1, 2$  which are known in target tracking as zero-order derivative models, white noise acceleration models, and Wiener process acceleration models, respectively.

---

**Algorithm 3** Tracking of the time-varying underlying distribution
 

---

**Input:**  $(x_{t-1}, y_{t-1}), \hat{\boldsymbol{\eta}}_{t-1,i}, \hat{\boldsymbol{\Sigma}}_{t-1,i}, \hat{p}_t(y = j), \mathbf{Q}_{t,i}$ , and  $r_{t,i}$  for  $j = 1, 2, \dots, |\mathcal{Y}|$  and  $i = 1, 2, \dots, m$

**Output:**  $\hat{\boldsymbol{\tau}}_t, \boldsymbol{\lambda}_t, \hat{\boldsymbol{\eta}}_{t,i}$ , and  $\hat{\boldsymbol{\Sigma}}_{t,i}$  for  $i = 1, 2, \dots, m$

**for**  $i = 1, 2, \dots, m$  **do**

Obtain state vector  $\hat{\boldsymbol{\eta}}_{t,i}$  using (14) and MSE  $\hat{\boldsymbol{\Sigma}}_{t,i}$  using (15)

Obtain the  $i$ -th component of mean vector estimate  $\hat{\boldsymbol{\tau}}_{t,i}$

using (17) and confidence vector  $\boldsymbol{\lambda}_{t,i}$  using (18)

---

## 5. Efficient learning of AMRCs

This section describes the proposed algorithms that obtain AMRCs' parameters and minimax risks at each time. First, we describe the accelerated subgradient method (ASM) that solves the convex optimization problem (7) using Nesterov extrapolation strategy (Nesterov & Shikhman, 2015; Tao et al., 2019). Then, we propose efficient algorithms that use a warm-start for the ASM iterations and maintain a local approximation of the polyhedral function  $\varphi(\cdot)$  in (6).

In what follows, we present techniques that efficiently obtain the classifier parameter  $\boldsymbol{\mu}_t$  and minimax risk  $R(\mathcal{U}_t)$  at time  $t$  from the classifier parameter  $\boldsymbol{\mu}_{t-1}$  and the updated mean vector estimate  $\hat{\boldsymbol{\tau}}_t$  and confidence vector  $\boldsymbol{\lambda}_t$ . The ASM algorithm applied to optimization (7) obtains classi-

fier parameters using the iterations for  $l = 1, 2, \dots, K$

$$\begin{aligned}\bar{\boldsymbol{\mu}}_t^{(l+1)} &= \boldsymbol{\mu}_t^{(l)} + a_l \left( \hat{\boldsymbol{\tau}}_t - \partial\varphi(\boldsymbol{\mu}_t^{(l)}) - \boldsymbol{\lambda}_t \odot \text{sign}(\boldsymbol{\mu}_t^{(l)}) \right) \\ \boldsymbol{\mu}_t^{(l+1)} &= \bar{\boldsymbol{\mu}}_t^{(l+1)} + \theta_{l+1}(\theta_l^{-1} - 1) \left( \boldsymbol{\mu}_t^{(l)} - \bar{\boldsymbol{\mu}}_t^{(l)} \right)\end{aligned}\quad (19)$$

where  $\boldsymbol{\mu}_t^{(l)}$  is the  $l$ -th iterate for  $\boldsymbol{\mu}_t$ ,  $\theta_l = 2/(l+1)$  and  $a_l = 1/(l+1)^{3/2}$  are step sizes, and  $\partial\varphi(\boldsymbol{\mu}_t^{(l)})$  denotes a subgradient of  $\varphi(\cdot)$  at  $\boldsymbol{\mu}_t^{(l)}$ .

The proposed algorithm reduces the number of ASM iterations by using a warm-start that initializes the parameters  $\boldsymbol{\mu}_t$  in (19) with the solution obtained at previous time  $\boldsymbol{\mu}_{t-1}$ . In addition, the ASM iterations are efficiently computed by maintaining a local approximation of the polyhedral function  $\varphi(\cdot)$  in (6). Such function is given by the pointwise maximum of linear functions indexed by pairs of instances and labels' subsets  $x \in \mathcal{X}, \mathcal{C} \subseteq \mathcal{Y}$ . So that, if  $\mathcal{I}$  is the set of such pairs we have that  $\varphi(\boldsymbol{\mu})$  in (6) becomes

$$\varphi(\boldsymbol{\mu}) = \max_{i \in \mathcal{I}} \{ \mathbf{f}_i^T \boldsymbol{\mu} - h_i \} \quad (20)$$

with  $\mathbf{f}_i = \sum_{y \in \mathcal{C}} \Phi(x, y)/|\mathcal{C}| \in \mathbb{R}^m$  and  $h_i = 1/|\mathcal{C}|$  for index  $i$  that corresponds to pair  $(x, \mathcal{C})$ . We use local approximations of (20) given by indices corresponding with the  $N$  most recent subgradients of  $\varphi(\cdot)$ . Specifically, if  $\mathcal{R} \subseteq \mathbb{R}^m$  we have that for any  $\boldsymbol{\mu} \in \mathcal{R}$ ,

$$\varphi(\boldsymbol{\mu}) = \max_{i \in \mathcal{J}_{\mathcal{R}}} \{ \mathbf{f}_i^T \boldsymbol{\mu} - h_i \}$$

with  $\mathcal{J}_{\mathcal{R}} = \{i \in \mathcal{I} : \exists \boldsymbol{\mu} \in \mathcal{R} \text{ with } \varphi(\boldsymbol{\mu}) = \mathbf{f}_i^T \boldsymbol{\mu} - h_i\}$   
 $= \{i \in \mathcal{I} : \exists \boldsymbol{\mu} \in \mathcal{R} \text{ with } \mathbf{f}_i \in \partial\varphi(\boldsymbol{\mu})\}$ .

Then, if  $\mathcal{J}_{\mathcal{R}} = \{i_1, i_2, \dots, i_N\}$ , we have that for any  $\boldsymbol{\mu} \in \mathcal{R}$ ,  $\varphi(\boldsymbol{\mu}) = \max\{\mathbf{F}\boldsymbol{\mu} - \mathbf{h}\}$  with  $\mathbf{F} = [\mathbf{f}_{i_1}, \mathbf{f}_{i_2}, \dots, \mathbf{f}_{i_N}]^T$  and  $\mathbf{h} = [h_{i_1}, h_{i_2}, \dots, h_{i_N}]^T$ .

Algorithm 4 details the proposed procedure to learn the classifier parameters  $\boldsymbol{\mu}_t$  together with the minimax risk  $R(\mathcal{U}_t)$  that provides the performance guarantees. Such algorithm has computational complexity  $O(NKm+m^2)$  and memory complexity  $O(Nm)$  where  $m = |\mathcal{Y}|d$  is the length of the feature mapping,  $N$  is the number of subgradients, and  $K$  is the number of iterations for ASM in (19). Note that the complexity of Algorithms 3 and 4 does not depend on the number of instance-label pairs and time steps. Therefore, the proposed algorithms are applicable to large-scale datasets and have constant complexity per time step.

## 6. Numerical Results

This section evaluates the performance of AMRCs in comparison with the presented performance guarantees and the state-of-the-art. In the first set of numerical results, we use

**Algorithm 4** Optimization for ARMC params. and minimax risk

**Input:**  $\hat{\boldsymbol{\tau}}_t, \boldsymbol{\lambda}_t, \boldsymbol{\mu}_{t-1}, x_{t-1}, \mathbf{F}$ , and  $\mathbf{h}$

**Output:**  $\boldsymbol{\mu}_t, R(\mathcal{U}_t), \mathbf{F}_{\text{new}}$ , and  $\mathbf{h}_{\text{new}}$

**for**  $\mathcal{C} \subseteq \mathcal{Y}, \mathcal{C} \neq \emptyset$  **do**

$\mathbf{F}, \mathbf{h} \leftarrow$  append rows  $\sum_{y \in \mathcal{C}} \Phi(x_{t-1}, y)^T/|\mathcal{C}|, 1/|\mathcal{C}|$  to  $\mathbf{F}, \mathbf{h}$

$\boldsymbol{\mu}_t^{(1)} \leftarrow \boldsymbol{\mu}_{t-1}, \bar{\boldsymbol{\mu}}_t^{(1)} \leftarrow \boldsymbol{\mu}_{t-1}$

**for**  $l = 1, 2, \dots, K$  **do**

$a_l \leftarrow 1/(l+1)^{3/2}, \theta_l \leftarrow 2/(l+1), \theta_{l+1} \leftarrow 2/(l+2)$

$\mathbf{f}_i^T \leftarrow$  row of  $\mathbf{F}$  such that  $\mathbf{f}_i^T \boldsymbol{\mu}_t^{(l)} - h_i = \max\{\mathbf{F}\boldsymbol{\mu}_t^{(l)} - \mathbf{h}\}$

$\bar{\boldsymbol{\mu}}_t^{(l+1)} \leftarrow \boldsymbol{\mu}_t^{(l)} + a_l \left( \hat{\boldsymbol{\tau}}_t - \mathbf{f}_i - \boldsymbol{\lambda}_t \odot \text{sign}(\boldsymbol{\mu}_t^{(l)}) \right)$

$\boldsymbol{\mu}_t^{(l+1)} \leftarrow \bar{\boldsymbol{\mu}}_t^{(l+1)} + \theta_{l+1}(\theta_l^{-1} - 1) \left( \boldsymbol{\mu}_t^{(l)} - \bar{\boldsymbol{\mu}}_t^{(l)} \right)$

$\mathbf{F}_{\text{new}}, \mathbf{h}_{\text{new}} \leftarrow$  append rows  $\mathbf{f}_i^T, h_i$  to  $\mathbf{F}_{\text{new}}, \mathbf{h}_{\text{new}}$

$\boldsymbol{\mu}_t \leftarrow \boldsymbol{\mu}_t^{(K+1)}$

$R(\mathcal{U}_t) \leftarrow 1 - \hat{\boldsymbol{\tau}}_t^T \boldsymbol{\mu}_t + \max\{\mathbf{F}\boldsymbol{\mu}_t - \mathbf{h}\} + \boldsymbol{\lambda}_t^T |\boldsymbol{\mu}_t|$

$\mathbf{F}_{\text{new}}, \mathbf{h}_{\text{new}} \leftarrow$  take the  $N$  most recent  $\mathbf{f}_i, h_i$  from  $\mathbf{F}_{\text{new}}, \mathbf{h}_{\text{new}}$

a synthetic dataset, while in the second set of numerical results, we use multiple benchmark datasets. The implementation of the proposed AMRCs is publicly available in Python and Matlab languages.<sup>1</sup> In addition, the appendices provide the detailed description of the benchmark datasets, additional implementation details, and supplementary numerical results.

We utilize a type of synthetic dataset that has been often used as benchmark for supervised classification under concept drift (Kumagai & Iwata, 2016; Kumagai & Iwata, 2017). At each time  $t$ , we generate a label  $y_t \in \{1, 2\}$  according to a Bernoulli distribution with parameter  $1/2$  and an instance given by

$$\begin{aligned}\mathbf{x}_t &= \left[ 4 \cos \left( \pi \left( (\cos(\omega t) - 3)/2 + y_t \right) \right) + \epsilon_1, \right. \\ &\quad \left. 4 \sin \left( \pi \left( (\cos(\omega t) - 3)/2 + y_t \right) \right) + \epsilon_2 \right]^T\end{aligned}$$

with  $\epsilon_1, \epsilon_2 \sim N(0, 2)$  and  $\omega = 0.1$ . In this paper we use a sinusoidal argument for cosines and sines so that the dataset is even more challenging. The class-conditional underlying distributions  $p_t(\mathbf{x}_t|y_t = 1)$  and  $p_t(\mathbf{x}_t|y_t = 2)$  are Gaussian with means that move with varying velocity and direction in a circle centered at the origin.

AMRCs are compared with the state-of-the-art using 12 datasets that have been often used as benchmarks for supervised classification under concept drift (Pavlidis et al., 2011; Kumagai & Iwata, 2016; Lu et al., 2016; Nguyen et al., 2017; Webb et al., 2018): ‘‘Weather’’, ‘‘Elec2’’, ‘‘Airlines’’, ‘‘German’’, ‘‘Chess’’, ‘‘Usenet1’’, ‘‘Usenet2’’, ‘‘Email Spam’’, ‘‘Credit card’’, ‘‘Smart grid sta-

<sup>1</sup><https://github.com/MachineLearningBCAM/AMRC-for-concept-drift-ICML-2022>

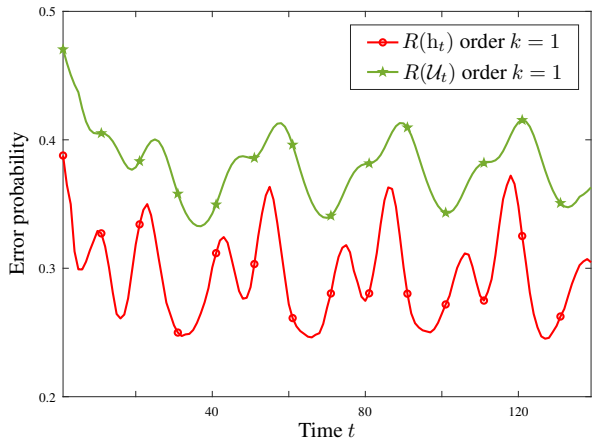


Figure 3. Results on synthetic data show the evolution in time of instantaneous performance bounds and error probabilities.

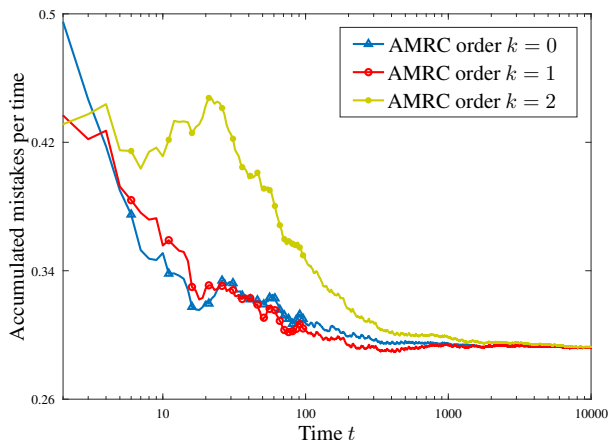


Figure 4. Results on synthetic data show the evolution in time of accumulated mistakes per number of steps.

bility”, “Shuttle”, and “Poker”. The last 2 datasets are multi-class problems and the rest are binary (see further details in the supplementary materials). The benchmark datasets can be obtained from UCI repository and from the Massive On-line Analysis library (Bifet et al., 2010).

The results for AMRCs are obtained using the feature mapping described in equation (1) of Section 2. Specifically, for the synthetic dataset we use the linear map  $\Psi(x) = x$  and for the benchmark datasets we use RFFs with  $D = 200$  as given by (2). In addition, we use the recursive approach presented in Akhlaghi et al., 2017 to obtain the variances  $\mathbf{Q}_{t,i}$  and  $r_{t,i}^2$  of noise processes  $\mathbf{w}_{t,i}$  and  $v_{t,i}$ , in (13); we obtain the probability of the labels using (12) with  $W = 200$ ; and the ASM in (19) is implemented with  $N = 100$  and  $K = 2000$ . AMRCs are compared with 10 state-of-the-art techniques: AdaRaker (Shen et al., 2019), randomized budget perceptron (RBP) (Cavallanti et al., 2007), Projectron, Projectron ++ (Orabona et al., 2008), naive online regularized risk minimization algorithm (NORMA) (Kivinen et al., 2004), Nyström online gradient descent (NOGD), Fourier

online gradient descent (FOGD) (Lu et al., 2016),  $\lambda$ -perceptron (Pavlidis et al., 2011), dynamic weighted majority (DWM) (Kolter & Maloof, 2007), and Forgetron (Dekel et al., 2006). The results of the 9 methods that use kernels utilize a scaling parameter calculated with a two-stage five-fold cross validation.

In the first set of numerical results we show the reliability of the presented instantaneous bounds using the synthetic data since the error probability in each time step cannot be computed using real-world datasets. Figure 3 shows the averaged instantaneous bounds of error probabilities corresponding to first inequality in (10) for  $\alpha_t = 0$  in comparison with the true error probabilities  $R(h_t)$  at each time. Such figure shows that the instantaneous bounds given by  $R(U_t)$  can offer tight upper bounds for the error probability at each time. In addition, Figure 4 shows the accumulated mistakes per time step of AMRCs of order  $k = 0, 1$ , and 2. As can be seen in such figure, an increased order can result in an improved overall performance at the expenses of worse initial performance.

In the second set of numerical results we use benchmark datasets to quantify AMRCs performance with respect to the state-of-the-art, the improvement due to multidimensional adaptation, and the reliability of the presented mistake bounds. Table 1 shows the classification error of AMRCs of order  $k = 1$  and the 10 state-of-the-art techniques for the 12 benchmark datasets described above.<sup>2</sup> The table shows that AMRCs and deterministic AMRCs (Det. AMRC) achieve the best performance in 3 and 9 datasets, respectively, and are competitive in all datasets. In addition, the running time of AMRCs is of the order of tens of milliseconds per time step similarly to the state-of-the-art techniques (see further details in Appendix C).

Table 1 shows that AMRCs offer an overall improved performance compared to the state-of-the-art along the benchmark datasets. In order to more clearly assess the performance improvement due to the multidimensional adaptation, we also implement a simple version of deterministic AMRCs that only account for a scalar rate of change (Unidim. AMRCs). Such AMRCs are obtained by using, for all the components  $i = 1, 2, \dots, m$ , the same gain in (14) and (15) that is taken as the average of the gains in (16). As shown in the table, the presented multidimensional adaptation can enable significant performance improvements in most datasets.

Figures 5 and 6 provide more detailed comparisons using the 3 techniques that have the top average rank (Det. AMRC, AMRC, and DWM) in “German” and “Usenet1” datasets. In addition, Figures 5 and 6 show the accumulated

<sup>2</sup>Bold numbers indicate the top result and N/As indicates that the classification method cannot be used with non-binary datasets.



Table 1. Classification error in % of AMRCs in comparison with state-of-the-art techniques.

Algorithm	Weather	Elec2	Airlines	German	Chess	Usenet1	Usenet2	Email	C. card	S. Grid	Shuttle	Poker	Ave. rank
AdaRaker	34.8	49.5	42.7	41.0	46.6	50.0	49.0	45.7	0.33	46.5	N/A	N/A	10.9
RBP	34.3	39.6	42.7	40.5	36.5	44.9	48.3	45.3	0.33	38.6	N/A	N/A	8.0
Projectron	30.7	36.8	41.8	40.0	35.1	49.1	47.7	48.2	0.33	<b>34.4</b>	64.7	22.6	6.2
Projectron ++	32.0	48.1	41.9	42.2	43.4	47.2	45.2	48.1	0.33	35.8	64.9	23.4	8.3
NORMA	35.5	43.2	40.7	39.7	39.4	36.0	32.8	34.6	0.18	38.8	N/A	N/A	6.6
NOGD	38.2	48.2	39.1	37.1	39.0	34.6	33.4	36.1	0.44	38.7	69.3	23.4	7.4
FOGD	32.0	44.9	41.1	37.6	38.3	45.9	47.0	44.9	0.33	36.0	64.9	22.9	6.4
$\lambda$ -perceptron	31.4	40.4	44.5	42.9	41.6	44.5	31.4	46.8	0.18	35.8	N/A	N/A	7.3
DWM	<b>30.0</b>	36.3	<b>37.8</b>	41.2	35.2	36.3	<b>28.8</b>	39.5	0.80	36.3	16.9	22.0	4.8
Forgetron	32.0	46.0	42.0	40.1	33.1	46.8	47.2	41.7	0.33	45.5	64.4	25.6	7.3
Unidim. AMRC	31.3	40.1	44.5	30.3	38.3	46.3	33.3	48.4	0.18	36.2	70.4	39.4	7.2
AMRC	32.3	35.8	38.9	30.3	<b>27.7</b>	35.7	30.9	43.7	<b>0.17</b>	35.8	15.2	<b>21.9</b>	3.0
Det. AMRC	<b>30.0</b>	<b>33.9</b>	39.4	<b>30.0</b>	33.4	<b>32.0</b>	29.9	<b>33.9</b>	<b>0.17</b>	<b>34.4</b>	<b>10.6</b>	<b>21.9</b>	<b>1.5</b>

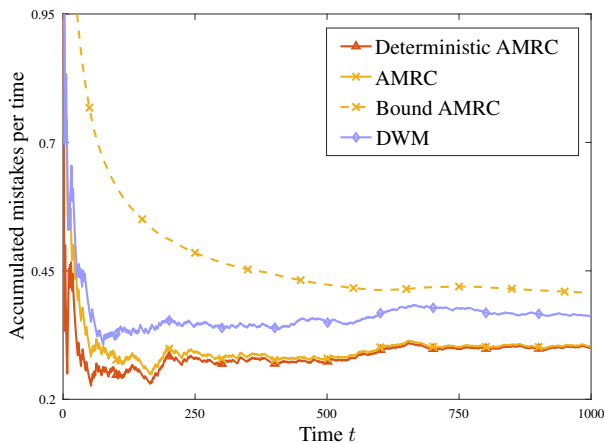


Figure 5: Results on “German” dataset shows the evolution of accumulated mistake bounds and accumulated mistakes per number of steps.

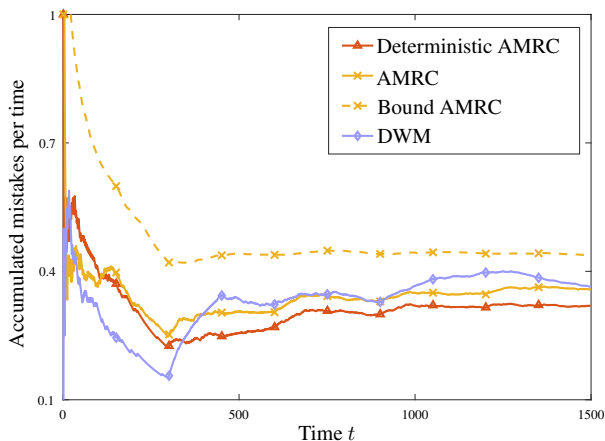


Figure 6: Results on “Usenet1” dataset shows the evolution of accumulated mistake bounds and accumulated mistakes per number of steps.

mistakes per number of steps  $\sum_{t=1}^T \mathbb{1} \{ \hat{y}_t \neq y_t \} / T$  at each time in comparison with the accumulated mistake bounds per number of steps corresponding to the first inequality in (11) for  $\delta = 0.05$  and  $\alpha_t = 0$ . As can be seen in such figure, the accumulated mistake bounds in (11) obtained at AMRC learning can offer accurate estimates for the classification error. These results together with those in Figure 3 show that the proposed methods can provide tight performance guarantees both in terms of instantaneous bounds for error probabilities and also in terms of bounds for accumulated mistakes.

## 7. Conclusion

The paper proposes the methodology of adaptive minimax risk classifiers (AMRCs) with multidimensional adaptation and performance guarantees. AMRCs learn classification rules estimating multiple statistical characteristics of the time-varying underlying distribution. In addition, AMRCs provide both qualitative and computable tight performance guarantees. The paper also proposes algorithms to track the time-varying underlying distribution accounting for multivariate and high-order time changes, and algorithms to efficiently update classification rules as new instance-label pairs arrive. The numerical results assess the performance improvement of AMRCs with respect to the state-of-the-art using benchmark datasets. The proposed methodology enables to more fully exploit data with characteristics that change over time and can provide more informative performance guarantees under concept drift.

## Acknowledgments

Funding in direct support of this work has been provided by the Spanish Ministry of Science, Innovation and Universities through Ramon y Cajal Grant RYC-2016-19383, BCAM’s Severo Ochoa Excellence Accreditation SEV-2017-0718, and Project PID2019-105058GA-I00, the Spanish Ministry of Economic Affairs and Digital Transformation through Project IA4TES MIA.2021.M04.0008, and by the Basque Government through the ELKARTEK, IT-1504-22, and BERC 2022-2025 programmes.

## References

- Akhlaghi, S., Zhou, N., and Huang, Z. Adaptive adjustment of noise covariance in Kalman filter for dynamic state estimation. In *IEEE Power & Energy Society General Meeting*, pp. 1–5, 2017.
- Bar Shalom, Y., Li, X. R., and Kirubarajan, T. *Estimation with applications to tracking and navigation: theory algorithms and software*. John Wiley & Sons, 2004.
- Bifet, A. and Gavaldà, R. Learning from time-changing data with adaptive windowing. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pp. 443–448. SIAM, 2007.
- Bifet, A., Holmes, G., Pfahringer, B., Kranen, P., Kremer, H., Jansen, T., and Seidl, T. MOA: Massive online analysis, a framework for stream classification and clustering. In *Proceedings of the First Workshop on Applications of Pattern Analysis*, pp. 44–50. PMLR, 2010.
- Cavallanti, G., Cesa Bianchi, N., and Gentile, C. Tracking the best hyperplane with a simple budget perceptron. *Machine Learning*, 69(2-3):143–167, 2007.
- Cutkosky, A. Parameter-free, dynamic, and strongly-adaptive online learning. In *International Conference on Machine Learning*, pp. 2250–2259. PMLR, 2020.
- Dekel, O., Shalev Shwartz, S., and Singer, Y. The Forgetron: A kernel-based perceptron on a fixed budget. In *Advances in Neural Information Processing Systems*, pp. 259–266, 2006.
- Delany, S. J., Cunningham, P., Tsymbal, A., and Coyle, L. A case-based technique for tracking concept drift in spam filtering. *Knowledge-Based Systems*, 4(18):187–195, 2005.
- Ditzler, G., Roveri, M., Alippi, C., and Polikar, R. Learning in nonstationary environments: A survey. *IEEE Computational Intelligence Magazine*, 10(4):12–25, 2015.
- Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4):1–37, 2014.
- Humpherys, J., Redd, P., and West, J. A fresh look at the Kalman filter. *SIAM review*, 54(4):801–823, 2012.
- Kivinen, J., Smola, A. J., and Williamson, R. C. Online learning with kernels. *IEEE Transactions on Signal Processing*, 52(8):2165–2176, 2004.
- Klinkenberg, R. Learning drifting concepts: Example selection vs. example weighting. *Intelligent data analysis*, 8(3):281–300, 2004.
- Kolter, J. Z. and Maloof, M. A. Dynamic weighted majority: An ensemble method for drifting concepts. *Journal of Machine Learning Research*, 8(12):2755–2790, 2007.
- Kumagai, A. and Iwata, T. Learning future classifiers without additional data. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 1772–1778, 2016.
- Kumagai, A. and Iwata, T. Learning non-linear dynamics of decision boundaries for maintaining classification performance. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2017.
- Long, P. M. The complexity of learning according to two models of a drifting environment. *Machine Learning*, 37(3):337–354, 1999.
- Lu, J., Hoi, S. C., Wang, J., Zhao, P., and Liu, Z.-Y. Large scale online kernel learning. *The Journal of Machine Learning Research*, 17(1):1613–1655, 2016.
- Mazuelas, S., Zanoni, A., and Perez, A. Minimax classification with 0-1 loss and performance guarantees. In *Advances in Neural Information Processing Systems*, pp. 302–312, 2020.
- Mazuelas, S., Shen, Y., and Pérez, A. Generalized maximum entropy for supervised classification. *IEEE Transactions on Information Theory*, 68(4):2530–2550, 2022a.
- Mazuelas, S., Romero, M., and Grünwald, P. Minimax risk classifiers with 0-1 loss. *arXiv preprint*, arXiv:2201.06487, 2022b.
- Mohri, M. and Medina, A. M. New analysis and algorithm for learning with drifting distributions. In *International Conference on Algorithmic Learning Theory*, pp. 124–138. Springer, 2012.
- Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.
- Namkoong, H. and Duchi, J. C. Variance-based regularization with convex objectives. *Advances in neural information processing systems*, 30, 2017.
- Nesterov, Y. and Shikhman, V. Quasi-monotone subgradient methods for nonsmooth convex minimization. *Journal of Optimization Theory and Applications*, 165(3): 917–940, 2015.
- Nguyen, T. D., Le, T., Bui, H., and Phung, D. Large-scale online kernel learning with random feature reparameterization. In *International Joint Conference on Artificial Intelligence*, pp. 2543–2549, 2017.

- Nguyen, T. T. T., Nguyen, T. T., Liew, A. W.-C., and Wang, S.-L. Variational inference based bayes online classifiers with concept drift adaptation. *Pattern Recognition*, 81: 280–293, 2018.
- Odelson, B. J., Rajamani, M. R., and Rawlings, J. B. A new autocovariance least-squares method for estimating noise covariances. *Automatica*, 42(2):303–308, 2006.
- Orabona, F., Keshet, J., and Caputo, B. The projectron: a bounded kernel-based perceptron. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 720–727, 2008.
- Pavlidis, N. G., Tasoulis, D. K., Adams, N. M., and Hand, D. J.  $\lambda$ -perceptron: An adaptive classifier for data streams. *Pattern Recognition*, 44(1):78–96, 2011.
- Shafieezadeh-Abadeh, S., Kuhn, D., and Esfahani, P. M. Regularization via mass transportation. *Journal of Machine Learning Research*, 20(103):1–68, 2019.
- Shen, Y., Chen, T., and Giannakis, G. B. Random feature-based online multi-kernel learning in environments with unknown dynamics. *The Journal of Machine Learning Research*, 20(1):773–808, 2019.
- Tao, W., Pan, Z., Wu, G., and Tao, Q. The strength of Nesterov’s extrapolation in the individual convergence of nonsmooth optimization. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2557–2568, 2019.
- Wang, H., Fan, W., Yu, P. S., and Han, J. Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 226–235, 2003.
- Webb, G. I., Lee, L. K., Goethals, B., and Petitjean, F. Analyzing concept drift and shift from sample data. *Data Mining and Knowledge Discovery*, 32(5):1179–1199, 2018.
- Zhang, L., Lu, S., and Zhou, Z.-H. Adaptive online learning in dynamic environments. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 1330–1340, 2018.

## Appendix A. Proof of Theorem 1.

*Proof.* For the first bound in (10), we have that

$$R(h_t) \leq \max_{p \in \mathcal{U}_t^\infty} \ell(h_t, p)$$

since  $R(h_t) = \ell(h_t, p_t)$  and  $p_t \in \mathcal{U}_t^\infty$  by definition of  $\mathcal{U}_t^\infty$ . Then, the maximization

$$\max_{p \in \mathcal{U}_t^\infty} \ell(h_t, p)$$

is obtained by using a slight reformulation of the optimization problem in the Theorem 2 in Mazuelas et al., 2020. Specifically, with the notation in this paper such optimization problem is

$$\begin{aligned} \min_{\mu, \nu} \quad & -\tau_t^T \mu - \nu \\ \text{s. t.} \quad & \Phi(x, y)^T \mu + \nu \leq h_t(y|x) - 1 \\ & \text{for any } (x, y) \in \mathcal{X} \times \mathcal{Y} \end{aligned}$$

In addition, the constraint of the above optimization problem can be rewritten as

$$\begin{aligned} \Phi(x, y)^T \mu + \nu &\leq h_t(y|x) - 1, \quad \forall x, y \\ \Leftrightarrow \nu &\leq h_t(y|x) - 1 - \Phi(x, y)^T \mu, \quad \forall x, y \\ \Leftrightarrow \nu &\leq - \max_{x \in \mathcal{X}, y \in \mathcal{Y}} \{\Phi(x, y)^T \mu - h_t(y|x) + 1\}. \end{aligned}$$

Then, minimizing  $-\nu$ , we obtain the optimization problem

$$\begin{aligned} \max_{p \in \mathcal{U}_t^\infty} \ell(h_t, p) \\ = \min_{\mu} 1 - \tau_t^T \mu + \max_{x \in \mathcal{X}, y \in \mathcal{Y}} \{\Phi(x, y)^T \mu - h_t(y|x)\} \end{aligned}$$

that results in

$$\begin{aligned} \max_{p \in \mathcal{U}_t^\infty} \ell(h_t, p) \\ = \min_{\mu} 1 - \tau_t^T \mu + \max_{x \in \mathcal{X}, y \in \mathcal{Y}} \{\Phi(x, y)^T \mu - h_t(y|x)\} \\ \leq 1 - \tau_t^T \mu_t + \max_{x \in \mathcal{X}, y \in \mathcal{Y}} \{\Phi(x, y)^T \mu_t - h_t(y|x)\} \\ \leq 1 - \tau_t^T \mu_t + \varphi(\mu_t) \end{aligned}$$

where the last inequality is due to the definition of  $h_t$  and  $\varphi(\cdot)$  in (5) and (6), respectively. Then, we have that

$$\begin{aligned} R(h_t) &\leq 1 - \tau_t^T \mu_t + \varphi(\mu_t) \\ &= 1 - \tau_t^T \mu_t + \varphi(\mu_t) + \widehat{\tau}_t^T \mu_t - \widehat{\tau}_t^T \mu_t + \lambda_t^T |\mu_t| - \lambda_t^T |\mu_t| \\ &= R(\mathcal{U}_t) - \tau_t^T \mu_t + \widehat{\tau}_t^T \mu_t - \lambda_t^T |\mu_t| \end{aligned} \quad (21)$$

that leads to the first bound in (10) for  $\alpha_t = \|\tau_t - \widehat{\tau}_t\|_\infty \|\mu_t\|_1$  using Hölder's inequality. In addition, the minimax risk given by the optimization problem (7) satisfies

$$\begin{aligned} R(\mathcal{U}_t) &= \min_{\mu} 1 - \widehat{\tau}_t^T \mu + \varphi(\mu) + \lambda_t^T |\mu| \\ &\leq 1 - \widehat{\tau}_t^T \mu_t^\infty + \varphi(\mu_t^\infty) + \lambda_t^T |\mu_t^\infty| \\ &= R_t^\infty + (\tau_t - \widehat{\tau}_t)^T \mu_t^\infty + \lambda_t^T |\mu_t^\infty| \end{aligned} \quad (22)$$

that together with (21) leads to the second bound in (10) for  $\beta_t = (\|\tau_t - \widehat{\tau}_t\|_\infty + \|\lambda_t\|_\infty) \|\mu_t^\infty - \mu_t\|$  since

$$\begin{aligned} R(h_t) &\leq R(\mathcal{U}_t) - \tau_t^T \mu_t + \widehat{\tau}_t^T \mu_t - \lambda_t^T |\mu_t| \\ &\leq R_t^\infty + (\tau_t - \widehat{\tau}_t)^T (\mu_t^\infty - \mu_t) + \lambda_t^T (|\mu_t^\infty| - |\mu_t|) \\ &\leq R_t^\infty + (\tau_t - \widehat{\tau}_t)^T (\mu_t^\infty - \mu_t) + \lambda_t^T |\mu_t^\infty - \mu_t| \quad (23) \\ &\leq R_t^\infty + (\|\tau_t - \widehat{\tau}_t\|_\infty + \|\lambda_t\|_\infty) \|\mu_t^\infty - \mu_t\|_1 \end{aligned}$$

where (23) is obtained using the reverse triangle inequality since  $\lambda_t$  is positive.

For the first bound in (10) in the case that  $\lambda_t \succeq |\tau_t - \widehat{\tau}_t|$ , we have that  $R(h_t) \leq R(\mathcal{U}_t)$  because  $p_t \in \mathcal{U}_t$  by definition of  $\mathcal{U}_t$ . Then, the minimax risk given by the optimization problem (7) satisfies (22) that leads to (10) for  $\alpha_t = 0$  and  $\beta_t = 2 \|\lambda_t\|_\infty \|\mu_t^\infty\|_1$  since  $\lambda_t \succeq |\tau_t - \widehat{\tau}_t|$ .

For the result in (11), let  $(x_1, y_1), (x_2, y_2), \dots, (x_T, y_T)$  be a sequence of instance-label pairs. If

$$V_t = \mathbb{1}\{\hat{y}_t \neq y_t\} - R(h_t)$$

we have that the sequence  $V_1, V_2, \dots$  is a martingale difference with respect to  $(x_1, y_1), (x_2, y_2), \dots$  because

$$\mathbb{E}[V_t | (x_1, y_1), (x_2, y_2), \dots, (x_{t-1}, y_{t-1})] = 0$$

for any  $t$ . Then, using the Azuma's inequality (Mohri et al., 2018), since  $|V_t| \leq 1$  for any  $t$ , we have that

$$\sum_{t=1}^T V_t \leq \sqrt{2T \log \frac{1}{\delta}} \quad (24)$$

with probability at least  $1 - \delta$ . Then, the result is obtained using bound in (10) because with probability at least  $1 - \delta$  we have that

$$\sum_{t=1}^T \mathbb{1}\{\hat{y}_t \neq y_t\} \leq \sum_{t=1}^T R(h_t) + \sqrt{2T \log \frac{1}{\delta}}$$

using (24) and the definition of  $V_t$ .  $\square$

## Appendix B. Derivation of the dynamical system

Dynamical systems as that given by (13) can be derived using the  $k$ -th order Taylor expansion of the conditional expectations  $\gamma_{t,i} = \mathbb{E}_{p_t}\{\Psi_r(x)|y = j\}$  with  $i = (d-1)j + r$  for  $j = 1, 2, \dots, |\mathcal{Y}|$  and  $r = 1, 2, \dots, d$ . The Taylor expansions of  $\gamma_{t,i}$  and its successive derivatives up to order  $k$  are given by

$$\begin{aligned}\gamma_{t,i} &\approx \gamma_{t-1,i} + \Delta_t \gamma'_{t-1,i} + \frac{\Delta_t^2}{2} \gamma''_{t-1,i} + \dots + \frac{\Delta_t^k}{k!} \gamma^{(k)}_{t-1,i} \\ \gamma'_{t,i} &\approx \gamma'_{t-1,i} + \Delta_t \gamma''_{t-1,i} + \dots + \frac{\Delta_t^{k-1}}{(k-1)!} \gamma^{(k)}_{t-1,i} \\ &\vdots \\ \gamma^{(k)}_{t,i} &\approx \gamma^{(k)}_{t-1,i}\end{aligned}$$

where  $\Delta_t$  is the time increment at  $t$  and  $\gamma^{(k)}_{t,i}$  denotes the  $k$ -th derivative of  $\gamma_{t,i}$ . The above equations lead to

$$\boldsymbol{\eta}_{t,i} \approx \begin{pmatrix} 1 & \Delta_t & \frac{\Delta_t^2}{2} & \dots & \frac{\Delta_t^k}{k!} \\ 0 & 1 & \Delta_t & \dots & \frac{\Delta_t^{k-1}}{(k-1)!} \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{pmatrix} \boldsymbol{\eta}_{t-1,i} \quad (25)$$

since  $\boldsymbol{\eta}_{t,i}$  is composed by  $\gamma_{t,i}$  and its derivatives up to order  $k$ . In addition,  $\gamma_{t,i}$  is observed at each time  $t$  through the instance-label pair  $(x_t, y_t)$ , so that we have

$$\Phi_i(x_t, y_t) \approx \gamma_{t,i}, \quad \text{if } y_t = j \quad (26)$$

with  $i = (d-1)j + r$  for  $j = 1, 2, \dots, |\mathcal{Y}|$  and  $r = 1, 2, \dots, d$ . Then, equations (25) and (26) above lead to the dynamical system in (13).

## Appendix C. Implementation details and additional results

In this section we describe the implementation details used for the numerical results in Section 6 and show additional results. In addition, the Python and Matlab implementations of the proposed AMRCs are available on the web <https://github.com/MachineLearningBCAM/AMRC-for-concept-drift-ICML-2022> with the setting used in the numerical results.

In the synthetic dataset described in Section 6, the class-conditional underlying distributions  $p_t(\mathbf{x}_t|y_t = 1)$  and  $p_t(\mathbf{x}_t|y_t = 2)$  are Gaussian with means that move with varying velocity and direction in a circle centered at the origin. Specifically, the speed of such means has periodicity  $\pi/\omega$  with maxima at times  $t = ((n-1)\pi + \pi/2)/\omega$  and minima at times  $t = n\pi/\omega$ , for  $n \in \mathbb{N}$ . In addition, the direction of movement changes with the same periodicity at times when the velocity is minimum.

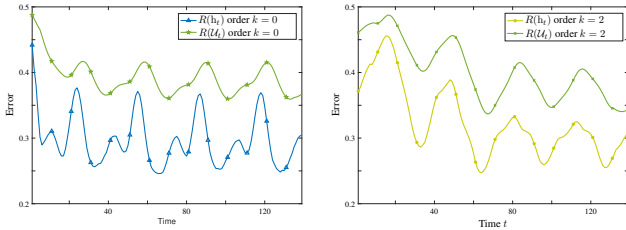
In the real-world datasets used in Section 6, we use 13 public datasets that have been often used as benchmark in adaptive settings. These datasets are further described in Table 2 that shows the number of instance-label pairs, the dimensionality of instances, and the number of labels. Three of such datasets are large-scale (“Airlines”, “Credit card”, and “Poker”), four of them are medium-sized datasets (“Weather”, “Phishing”, “Smart grid stability”, and “Shuttle”), and the rest are short-sized datasets (“Elec2”, “German”, “Chess”, “Usenet1”, “Usenet2”, and “Email”). For instance, the “Airlines” dataset used contains 539,383 instances with flight arrival and departure information and aims to predict if a flight will be delayed or not; the “Weather” dataset used contains 18,159 instances with daily measurements of weather factors and aims to predict if it will rain or not; and the “Elec2” dataset used contains 1,148 instances with twice-daily measurements (12am and 12pm) of factors that affect load demand and price and aims to predict if the price will be higher or not.

The results in Table 1 for methods that use kernels are obtained with a scaling parameter  $\gamma$  calculated using a two-stage five-fold cross validation. Specifically, at the first stage, the values for the scaling factor are selected from  $2^i$  for  $i = \{-6, -3, 0, 3, 6\}$ . At the second stage, if  $\gamma_0 = 2^i$  where  $i$  is the best parameter obtained at the first stage, then the values for the scaling parameters are selected from  $\gamma_0 2^i$ ,  $i = \{-2, -1, 0, 1, 2\}$ . The final value is  $\gamma = \gamma_0 2^i$  where  $i$  is the best parameter obtained at second stage.

In the first set of additional results we further illustrate the reliability of instantaneous bounds using the synthetic dataset described in Section 6. Specifically, we extend such results for AMRCs with orders  $k = 0$  and 2 completing those in the main paper. In order to quantify the error proba-

Table 2. Datasets information show number of instances, dimensionality of instances, and number of labels.

Dataset	Time Steps	Dim. of instances	$ \mathcal{Y} $
Weather	18,159	8	2
Elec2	1,148	4	2
Airlines	539,383	7	2
German	1,000	24	2
Chess	503	8	2
Usenet 1	1,500	99	2
Usenet 2	1,500	99	2
Email	1,498	913	2
C. card	284,806	30	2
S. Grid	60,000	13	2
Shuttle	100,001	9	4
Poker	829,201	10	10



(a). Instantaneous bounds and probabilities of error for AMRC of order  $k = 0$ . (b). Instantaneous bounds and probabilities of error for AMRC of order  $k = 2$ .

Figure 7. Results on synthetic data show the evolution in time of instantaneous bounds and probabilities of error.

Table 3. Averaged running times per time step in milliseconds of AMRC in comparison with state-of-the-art techniques.

Algorithm	Mean
RBP	9.2
Proj.	9.0
P. ++	9.1
NORMA	6.8
A.Raker	10.1
NOGD	4.6
FOGD	1.9
$\lambda$ -per.	12.2
DWM	6.6
Forg.	6.9
AMRC $K = 500$	14.4
AMRC $K = 2000$	51.9

 Table 4. Classification error in % of AMRC with  $K = 500$  and  $K = 2000$  iterations in ASM.

Dataset	AMRC	
	$K = 500$	$K = 2000$
Weather	32.4	32.3
Elec2	35.9	35.8
Airlines	39.2	38.9
German	30.4	30.3
Chess	28.8	27.7
Usenet1	36.8	35.7
Usenet2	30.6	30.9
Email	43.0	43.7
C. card	2.53	0.17
S. grid	36.3	35.8
Shuttle	44.6	15.2
Poker	30.4	21.9

bility at each time, we use 10,000 Monte Carlo simulations. Figures 7a and 7b show the averaged instantaneous bounds of probabilities of error in comparison with the true probabilities of error  $R(h_t)$  at each time for AMRCs with order  $k = 0$  and  $k = 2$ , respectively. Such figures show that the instantaneous bounds  $R(\mathcal{U}_t)$  obtained at AMRCs learning can offer accurate estimates for the probability of error at each time.

In the second set of additional results, we assess the running time of AMRCs. Table 3 shows the averaged running time in milliseconds of AMRCs and the 10 state-of-the-art techniques for the 12 benchmark datasets. Such table shows the running time of AMRCs using  $K = 500$  and  $K = 2000$  iterations in the ASM in Section 5, and Table 4 shows the classification error of AMRCs with  $K = 500$  and  $K = 2000$  completing those in the main paper. The running time of AMRCs is of the order of tens of milliseconds per time step similarly to state-of-the-art techniques.