







Classifying Image Series with a Reoccurring Concept Drift Using a Markov Chain Predictor as a Feedback

Magda Skoczeń , Wojciech Rafajłowicz , and Ewaryst Rafajłowicz  

Faculty of Electronics, Wrocław University of Science and Technology,
Wrocław, Poland

{magda.skoczen,wojciech.rafajlowicz,ewaryst.rafajlowicz}@pwr.edu.pl

Abstract. We consider the problem of classifying images, ordered in a sequence (series), to several classes. A distinguishing feature of this problem is the presence of a slowly varying, reoccurring concept drift which results in the existence of relatively long subsequences of similar images from the same class. It seems useful to incorporate this knowledge into a classifier. To this end, we propose a novel idea of constructing a Markov chain (MC) that predicts a class label for the next image to be recognized. Then, the feedback is used in order to modify slightly a priori probabilities of class memberships. In particular, the a priori probability of the class predicted by the MC is increased at the expense of decreasing the a priori probabilities of other classes.

The idea of applying an MC predictor of classes with the feedback to a priori probabilities is rather general. Thus, it can be applied not only to images but also to vectors of features arising in other applications. Additionally, this idea can be combined with any classifier that is able to take into account a priori probabilities. We provide an analysis when this approach leads to the reduction of the classification errors in comparison to classifying each image separately.

As a vehicle to elucidate the idea we selected a recently proposed classifier (see [23, 25]) that is based on assuming matrix normal distribution (MND) of classes. We use a specific structure of the MND's covariance matrix that can be estimated directly from images without feature extraction.

The proposed approach was tested on simulated images as well as on an example of the sequence of images from an industrial gas burner's flames. The aim of the classification is to decide whether natural gas contains a sufficient amount of methane (the combustion process is proper, lower air pollution) or not.

Keywords: Classification of ordered images · Reoccurring concept drift · Generalized matrix normal distribution · Classifier with feedback · Markov chain predictor

1 Introduction

A large number of examples can be pointed out when images should be considered as a series (i.e., a sequence of images that are ordered in time). This is of particular importance when we want to observe and classify selected features over time. We assume that the features of interest can fluctuate over a long time intervals without changing their class membership of subsequent images. At unknown time instants, the observed features can change (abruptly or gradually), leading to the change of their classification for a longer period. Then, the next parts of the series can return to previously visited class memberships. We shall name such sequences *image series with a slow reoccurring concept drift*. The assumption that we are faced with an image series with a slow reoccurring concept drift is crucial for this paper. The reason is that we can use this fact in order to increase the classification accuracy by predicting a class label of the next image and incorporating this a priori knowledge into a classifier.

The following examples of such image series can be pointed out.

- Quality control of manufacturing processes monitored by cameras. The following two cases can be distinguished:
 1. Separate items are produced and classified as proper or improper (conforming to requirements or not). One can expect long sequences of conforming items and shorter sequences of non-conforming ones.
 2. Production is running continuously in time (e.g., molding and rolling of copper, other metals or steel alloys). Here also slow changes of the quality can be expected, while the proper classification of a product is crucial for economic efficiency.
- Combustion processes monitored by a camera (see [20,29] and the bibliography cited therein).
- Automated Guided Vehicles (AGV) are more and more frequently monitored by cameras and/or other sensors. Most of the time they operate in an environment which is well defined and built-in into their algorithms, but sometimes it happens that they meet other AVG's or other obstacles to be recognized.

For simplicity of the exposition we impose the following constraints on an image series:

- it is not required that class labels of subsequent images strictly form a Markov chain (MC), but if it is possible to indicate the corresponding MC, then the proposed classification algorithm has a stronger theoretical justification,
- images are represented as matrices, i.e., only their grey levels are considered,
- an image series has slow reoccurring concept drift in the sense explained above,
- the probability density functions (p.d.f.'s) of images in each class have covariance structures similar to that of matrix normal distributions (MND's).

The last assumption will be stated more formally later. We usually do not have enough learning images to estimate the full covariance matrix of large images. An alternative approach, that is based projections is proposed in [26].

The paper is organized as follows:

- in the section that follows we provide a short review of the previous works,
- in Sect. 3, we state the problem more precisely,
- in the next sections, we provide a skeletal algorithm, then its exemplifications and the results of testing are discussed.
- an example of classifying flames of an industrial gas burner is discussed in the section before last.

Notice that we place an emphasis on classifying images forming a time series, but avoiding a tedious (and costly) features extraction step (an image is considered as an entity).

2 Previous Work

We start from papers on the bayesian classifiers that arise in cases when the assumption that class densities have the MND distribution holds. This assumption will be essentially weakened in this paper, but it cannot be completely ignored when one wants to avoid the feature extraction step without having millions of training and testing images.

Then, we briefly discuss classic papers on classifying images when their class labels form a Markov chain. We also indicate the differences in using this information in classic papers and here.

As mentioned in [25], “The role of multivariate normal distributions with the Kronecker product structure of the covariance matrix for deriving classifiers was appreciated in [15], where earlier results are cited. In this paper the motivation for assuming the Kronecker product structure comes from repeated observations of the same object to be classified.”

It was also documented in [22,23] and [24] that the classifier based on the MND’s assumption is sufficiently efficient for classifying images without features extraction. Furthermore, in [25] it was documented that an extended version of such classifiers is useful for recognizing image sequences, considered as whole entities.

The second ingredient that is used in the present paper is a Markov chain based predictor of class labels, which is used as feedback between the classifier output and a priori class probabilities for classifying the next image in a sequence. This idea was proposed by the first author (see Fig. 1 for a schematic view of the overall system).

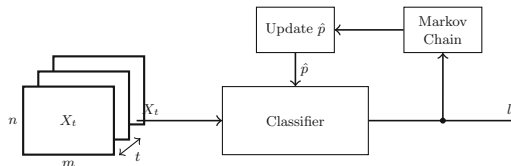


Fig. 1. A general scheme of classifying an image series.

The idea of using the assumption that class labels form a Markov structure has a long history [13] and its usefulness is appreciated up to now (see, e.g., [14, 28]). In these and many other papers, the Markov chain assumption was used in order to improve the classification accuracy of a present pattern using its links to the previous one. Here, we use a Markov chain differently, namely, it serves in the feedback as a predictor of the next class label.

A recurrence in neural networks has reappeared several times. A well-known class is Elman networks (see the original paper [9] and [4] for a learning process of Elman's networks which is so laborious that it requires parallel computing tools). In Elman's networks, an input vector is extended by stacking values of all the outputs from the previous iteration.

Hopfield neural networks (HNN) have even a more complicated structure. Their learning is also rather laborious, since after presenting an input vector the HNN internally iterates many times until a stable output is attained. We refer the reader to [18] for the description of the HNN that is dedicated to change detection between two images and to [2] for the HNN designed to classify temporal trajectories.

In the context of Elman's and Hopfield's nets, our feedback can be called weak, for the following reasons:

1. only limited information (class labels) is transmitted back to the recognizer,
2. the transmission is done once per image to be recognized (there are no inner loop iterations),
3. feedback information does not influence an input image, but provides only an outer context for its classification, by modifying a priori class probabilities.

The problem of dealing with reoccurring concept drift has been intensively studied in recent years. We refer the reader to [8, 10] and the bibliography cited therein. We adopt the following definition from [8]: "We say that the changes are reoccurring if we know that the concept can appear and disappear many times during processing the stream." We agree also with the following opinion: "Many methods designed to deal with concept-drift show unsatisfactory results in this case [3]."

Our approach can also be used for data streams, but after a learning phase that is based on long historical sequences of images. We shall not emphasise this aspect here. We refer the reader to [12] for the recent survey paper on learning for data stream analysis.

3 Problem Statement

In this section, we first recall the pattern recognition problem for matrices in the Bayesian setting. Then, we specify additional assumptions concerning a series of images and reoccurring concept drift.

3.1 Classic Bayesian Classification Problem for Random Matrices

A sequence of ordered images \mathbf{X}_k , $k = 1, 2, \dots, K$ is observed one by one. Grey level images are represented by $m \times n$ real-valued¹ matrices.

Each image \mathbf{X}_k , $k = 1, 2, \dots, K$ can be classified to one of $J > 1$ classes, labeled as $j = 1, 2, \dots, J$. The following assumptions apply to all J classes.

As 1) Each \mathbf{X}_k is a random matrix, which was drawn according to one of the probability density functions (p.d.f.'s), denoted further by $f_j(\mathbf{X})$, $j = 1, 2, \dots, J$, of the $m \times n$ matrix argument \mathbf{X}

As 2) Random matrices \mathbf{X}_k , $k = 1, 2, \dots, K$ are mutually independent (this does not exclude stochastic dependencies between elements of matrix \mathbf{X}_k).

As 3) Each matrix \mathbf{X}_k is associated with class label $j_k \in \mathcal{J}$, $k = 1, 2, \dots, K$, where $\mathcal{J} \stackrel{def}{=} \{j : 1 \leq j \leq J\}$.

Labels j_k 's are random variables, but – as opposed to the classic problem statement – they are not required to be mutually independent. When new matrix (image) \mathbf{X}_0 has to be classified to one of the classes from \mathcal{J} , then the corresponding label j_0 is not known. On the other hand, when ordered pairs (\mathbf{X}_k, j_k) , $k = 1, 2, \dots, K$ are considered as a learning sequence, then both images \mathbf{X}_k 's and the corresponding labels j_k 's are assumed to be known exactly, which means that a teacher (an expert) properly classified images \mathbf{X}_k 's.

As 4) We assume that for each class there exists a priori probability $p_j > 0$ that \mathbf{X} and \mathbf{X}_k 's were drawn from this class. Clearly $\sum_{j=1}^J p_j = 1$. Notice that here a priori probabilities are interpreted in the classic way, namely, as the probabilities drawing some of \mathbf{X}_k 's from j -th class. In other words, in specifying p_j 's we do not take into account the fact that \mathbf{X}_k 's form an image series with possible concept drift. Later, we modify this assumption appropriately.

As 5) (optional) In some parts of the paper we shall additionally assume that class densities $f_j(\mathbf{X})$'s have matrix normal distributions $\mathcal{N}_{n,m}(\mathbf{M}_j, U_j, V_j)$, i.e., with the expectation matrices \mathbf{M}_j and U_j as $n \times n$ inter-rows covariance matrices and V_j as an $m \times m$ covariance matrix between columns, respectively. We refer the reader to [23] and the bibliography cited therein for the definition and basic properties of MND's.

It is well-known (see, e.g., [7]) that for the 0–1 loss function the Bayes risk of classifying \mathbf{X} is minimized by the following classification rule:

$$j^* = \arg \max_{1 \leq j \leq J} p_j f_j(\mathbf{X}), \quad (1)$$

since a posteriori probabilities $P(j|\mathbf{X})$ of \mathbf{X} being from class $j \in \mathcal{J}$ are proportional to $p_j f_j(\mathbf{X})$, $j \in \mathcal{J}$ for each fixed \mathbf{X} .

In [23] particular cases of (1) were derived under assumption As 5).

¹ In implementations, grey levels are represented by integers from the range 0 to 255, but for theoretical considerations we consider them to be real numbers.

3.2 Classification of Images in a Series

Consider a discrete-time and discrete states Markov chain (MC) that is represented by $J \times J$ matrix \mathcal{P} with elements $p_{jj'} \geq 0$, $j, j' \in \mathcal{J}$, where $p_{jj'}$ is interpreted as the probability that the chain at state j moves to state j' at the next step. States of the MC are further interpreted as labels attached to \mathbf{X} 's. Clearly, \mathcal{P} must be such that $\sum_{j' \in \mathcal{J}} p_{jj'} = 1$ for each $j \in \mathcal{J}$. Additionally, the MC is equipped with column vector $\bar{q}_0 \in R^J$ with nonnegative elements that sum up to 1. The elements of this vector are interpreted as the probabilities that the MC starts from one of the states from \mathcal{J} . According to the rules governing Markov chains, vectors \bar{q}_k of the probabilities of the states in subsequent steps are given by $\bar{q}_k = \mathcal{P} \bar{q}_{k-1}$, $k = 1, 2, \dots, K$. At each step k a particular state of the MC is drawn at random according to the probabilities in \bar{q}_k .

As 6) Class labels j_k in series of images (\mathbf{X}_k, j_k) , $k = 1, 2, \dots, K$ are generated according to the above described Markov chain scheme.

In order to assure that the MC \mathcal{P} is able to model reoccurring concept drifts, represented by changes of j_k 's, we impose the following requirements on \mathcal{P} :

As 7) The MC \mathcal{P} possesses the following properties: all the states are communicating and recurrent, the MC is irreducible and aperiodic.

In the Introduction (see also the section before last) it was mentioned that our motivating example is the combustion process of natural gas, which can have slowly varying contents of methane modelled here as slowly varying, reoccurring concept drift. Similar concept drift changes can be observed in drinking water and the electricity supply networks in which concept drifts appear as changes of parameters of a supplied medium (e.g., the purity of water or the voltage of the current).

The ability of MC to generate long sequences of the same labels is guaranteed by the following assumption.

As 8) Transition matrix \mathcal{P} of the MC has strongly dominating diagonal elements (by strong dominance we mean the diagonal elements are about 5–10 times larger than other entries of \mathcal{P}).

Mimicking decision rule (1) for images in a series, under assumptions A1)-A4) and A6)-A8), it is expedient to consider the following sequence of decision rules:

$$j_k^* = \arg \max_{1 \leq j' \leq J} [p_{j_{(k-1)}^* j'} f_{j'}(\mathbf{X}_k)], \quad k = 1, 2, \dots, K, \quad (2)$$

where it is assumed that at step $(k-1)$ it was decided that j_{k-1}^* is the label maximizing the same expression (2), but at the previous step.

Under assumptions As 1) - As 4) and As 6) - As 8), our aim in this paper is the following:

1. having learning sequences of images: $(\hat{\mathbf{X}}_k, \hat{j}_k)$, $k = 1, 2, \dots, \hat{K}$
2. and assuming proper classifications \hat{j}_k to one of the classes
3. to construct an empirical classifier that is based on (2) decision rules.

and to test this rule on simulated and real data.

4 A General Idea of a Classifier with the MC Feedback

Decision rules (2) suggest the following empirical version of decision rules: \mathbf{X}'_k is classified to class \tilde{j}_k at k -th step if

$$\tilde{j}_k = \arg \max_{1 \leq j' \leq J} \left[\tilde{p}_{\tilde{j}_{(k-1)} j'} \hat{f}_{j'}(\mathbf{X}'_k) \right], \quad k = 1, 2, \dots, K, \quad (3)$$

where $\tilde{j}_{(k-1)}$ is the class indicated at the previous step, while:

1. $\hat{f}_{j'}(\mathbf{X}_k)$, $j' \in \mathcal{J}$ are estimators of p.d.f.'s $f_{j'}(\mathbf{X}_k)$ that are based on the learning sequence.
2. $\tilde{p}_{\tilde{j}_{(k-1)} j'}$, $j' \in \mathcal{J}$ plays the role of the a priori class probabilities at k -th step. Possible ways of updating them are discussed below.

As estimators in 1) one can select either

- parametric families with estimated parameters (an example will be given in the next section),
- nonparametric estimators, e.g., from the Parzen-Rosenblat family or apply another nonparametric classifier that allows specifying a priori class probabilities, e.g., the support vector machine (SVM) in which class densities are not explicitly estimated (see an example in the section before last).

The key point of this paper is the way of incorporating the fact that we are dealing with an image series with a reoccurring concept drift into the decision process. As already mentioned (see Fig. 1), the idea of a decision process is to modify a priori class probabilities taking into account the result of classifying the previous image by giving preferences (but not a certainty) to the latter class.

This idea can be formalized in a number of ways. We provide two of them that were tested and reported in this paper.

Skeletal classifier for image series with the MC feedback

Step 0 Initial learning phase: select estimators $\hat{f}_{j'}(\mathbf{X})$, $j' \in \mathcal{J}$ and calculate them from the learning sequence, estimate the original a priori class probabilities $\tilde{p}_{j'}^0$, $j' \in \mathcal{J}$ as the frequencies of their occurrence in the learning sequence. Then, set

$$\tilde{p}_{\tilde{j}_0 j'} = \tilde{p}_{j'}^0, \quad \text{for all } j' \in \mathcal{J} \quad (4)$$

for the compatibility of the notations in further iterations, where for $k = 0$ we select $\tilde{j}_0 = \arg \max_{1 \leq j \leq J} [\tilde{p}_j^0]$ as the maximum a priori probability. Estimate transition matrix $\tilde{\mathcal{P}}$ of the MC from the labels of the learning sequence. Select $\gamma \in [0, 1]$ and $\Delta_p \in [0, 1)$ as parameters of the algorithm (see remarks after the algorithm for explanations). Set $k = 1$ and go to Step 1.

Step 1 Decision: acquire the next image in the series \mathbf{X}'_k and classify it to class

$$\tilde{j}_k = \arg \max_{1 \leq j' \leq J} \left[\tilde{p}_{\tilde{j}(k-1)j'} \hat{f}_{j'}(\mathbf{X}'_k) \right]. \quad (5)$$

Step 2 Prediction: run one step ahead of the MC $\hat{\mathcal{P}}$, starting from the initial state: $[0, 0, \dots, \underbrace{1}_{\tilde{j}_k}, \dots, 0, 0]$. Denote by \check{j}_k the label resulting from this operation.

Step 3 Update the probabilities in (5) as follows:

$$\tilde{p}_{\tilde{j}_k j'} = \check{p}_{\tilde{j}_k j'} / \sum_{j=1}^J \check{p}_{\tilde{j}_k j} \text{ for all } j' \in \mathcal{J}, \quad (6)$$

where $\check{p}_{\tilde{j}_k j'}$'s are defined as follows: if $j' \neq \check{j}_k$, then for all other $j' \in \mathcal{J}$

$$\check{p}_{\tilde{j}_k j'} = (1 - \gamma) \check{p}_{j'}^0 + \gamma \tilde{p}_{\tilde{j}(k-1)j'}, \quad (7)$$

otherwise, i.e., for $j' = \check{j}_k$

$$\check{p}_{\tilde{j}_k j'} = (1 - \gamma) \check{p}_{j'}^0 + \gamma \tilde{p}_{\tilde{j}(k-1)j'} + \Delta_p. \quad (8)$$

Step 4 If \mathbf{X}'_k is not the last image in the series (or if \mathbf{X}'_k 's form a stream), then set $k := k + 1$ and go to Step 1, otherwise, stop.

The following remarks are in order.

1. In (7) and (8) parameter $\gamma \in [0, 1]$ influences the mixture of the original a priori information and information gathered during the classification of subsequent images from the series.
2. Parameter $\Delta_p \in [0, 1)$ is added only to the probability of the predicted class \check{j}_k in order to increase its influence on the next decision. The necessary normalization is done in (6). In the example presented in the section before last the influence of Δ_p on the classification accuracy is investigated.

5 Classifying Series of Images Having Matrix Normal Distributions

In this section, we exemplify and test the skeletal algorithm when As 5) is in force. In more details, for $m \times n$ images (matrices) \mathbf{X} class distributions have the MND's of the following form (see [17]):

$$f_j(\mathbf{X}) = \frac{1}{c_j} \exp \left[-\frac{1}{2} \text{tr} [U_j^{-1} (\mathbf{X} - \mathbf{M}_j) V_j^{-1} (\mathbf{X} - \mathbf{M}_j)^T] \right], \quad j \in \mathcal{J}, \quad (9)$$

where it is assumed that $n \times n$ matrices U_j 's and $m \times m$ matrices V_j 's are assumed to be nonsingular, while $n \times m$ matrices \mathbf{M}_j 's denote the class means. The normalization constants c_j 's are defined as follows:

$$c_j \stackrel{def}{=} (2\pi)^{0.5nm} \det[U_j]^{0.5n} \det[V_j]^{0.5m}, \quad j \in \mathcal{J}. \quad (10)$$

As one can observe, (9) has a special structure of the covariance matrix, namely, U_j 's are the covariance matrices between rows only, while V_j 's between columns only, which makes it possible to estimate them from learning sequences of reasonable lengths. This is in contrast to a general case when the full covariance matrix would have $mn \times mn$ elements, which is formidable for estimation even for small images $m = n = 100$. We refer the reader to [16, 27] for the method of estimating U_j 's and V_j 's from the learning sequence. Further on we denote these estimates by \hat{U}_j 's and \hat{V}_j 's, respectively. Analogously, we denote by $\hat{\mathbf{M}}_j$'s the estimates of the expectations of f_j 's that are obtained as the mean values of the corresponding matrices from the learning sequence, taking into account, however, that the entries of \mathbf{X}_k 's are contained either in $[0, 255]$ or $[0, 1]$, depending on the convention.

Matrix Classifier with a Markov Chain Feedback (MCLMCF)

Step 0: Select $0 < \kappa_{max} \leq 100$. Select $\Delta_p \in (0, 1)$.

Step 1: Initial learning phase: estimate $\hat{\mathbf{M}}_j, \hat{U}_j, \hat{V}_j, \hat{c}_j$ for $j \in \mathcal{J}$ from a learning sequence. Select $c_{min} = \min(\hat{c}_j)$, estimate the original a priori class probabilities $\tilde{p}_j^0, j \in \mathcal{J}$ as class occurrence frequencies in a learning sequence. Calculate $\tilde{j}_0 = \max(\tilde{p}_j^0)$. Set $k = 1$. Estimate $\hat{\mathcal{P}}$ based on the learning sequence. If $\hat{\mathcal{P}}$ has an absorbing state then use MCL [23]), otherwise go to step 2.

Step 2: Verify whether the following conditions hold:

$$\frac{\lambda_{max}(\hat{U}_j)}{\lambda_{min}(\hat{U}_j)} < \kappa_{max}, j \in \mathcal{J} \quad \text{and} \quad \frac{\lambda_{max}(\hat{V}_j)}{\lambda_{min}(\hat{V}_j)} < \kappa_{max}, j \in \mathcal{J} \quad (11)$$

If so, go to step 3a, otherwise, go to step 4.

Step 3a: Classify new matrix \mathbf{X}'_k to class, according to (12). Go to step 3b.

$$\tilde{j}_k = \underset{0 < j' \leq J}{\operatorname{argmin}} \left[\frac{1}{2} \operatorname{tr}[\hat{U}_{j'}^{-1} (\mathbf{X}'_k - \hat{\mathbf{M}}_{j'}) \hat{V}_{j'}^{-1} (\mathbf{X}'_k - \hat{\mathbf{M}}_{j'})^T] \right] \quad (12)$$

$$- \log(\tilde{p}_{\tilde{j}_k} / c_{min})$$

Step 3b: Update estimate of a priori probability \tilde{p} , using the following procedure:

- (a) Select class \tilde{j} , according to probabilities in \tilde{j}_k row of transition matrix $\hat{\mathcal{P}}$.
- (b) Set $\tilde{p}_{\tilde{j}_k \tilde{j}} = \tilde{p}_{\tilde{j}_k} / \tilde{p}_{\tilde{j}_k \tilde{j}}$.
- (c) For every $j' \neq \tilde{j}, j' \in \mathcal{J}$ check if $\tilde{p}_{\tilde{j}_k} - \frac{\Delta_p}{J-1} > 0$, if so then update $\tilde{p}_{\tilde{j}_k j'}$ and $\tilde{p}_{\tilde{j}_k \tilde{j}}$ according to (13), else $\tilde{p}_{\tilde{j}_k j'}$ and $\tilde{p}_{\tilde{j}_k \tilde{j}}$ remain unchanged.

$$\tilde{p}_{\tilde{j}_k j'} = \tilde{p}_{\tilde{j}_{(k-1)} j'} - \frac{\Delta_p}{J-1} \quad \text{and} \quad \tilde{p}_{\tilde{j}_k \tilde{j}} = \tilde{p}_{\tilde{j}_{k\tilde{j}}} + \frac{\Delta_p}{J-1} \quad (13)$$

Set $k = k + 1$. If $k \leq K$ then go to step 3a else stop.

Step 4: Classify new matrix \mathbf{X}'_k according to the nearest mean rule. Add the current result $(\mathbf{X}'_k, \tilde{j}_k)$ to the training set, then update the estimates of \tilde{p}_{j_0} , $\tilde{\mathbf{M}}_j$, \tilde{U}_j , \tilde{V}_j , \tilde{c}_j and select c_{min} for $j \in \mathcal{J}$ in the same manner, as described in step 1. Set $k = k + 1$. If $k \leq K$ then go to step 2, otherwise stop.

6 Experiments on Simulated Data

In this section, we present the results of experiments performed on simulated data for the evaluation of the proposed classifier performance. We generated 486 sequences, each consisting of 300 matrices. Matrices were randomly generated according to one of the three matrix normal distributions, which represents three classes. MNDs parameters are presented in Table 1, where J_8 is an 8×8 matrix of ones and D_8 is an 8×8 tridiagonal matrix with 2 at the main diagonal and 1 along the bands running above and below the main diagonal. The order of classes in the sequence was modeled by the MC with transition probability matrix \mathcal{P} (see Eq. (14)). In the conducted experiments, the data were divided into a training and testing set in the following manner: the first 100 matrices of a sequence were used as a training set for classifiers, the remaining matrices were used as a test set. Then Gaussian noise with ($\sigma = 0.05$) was added to matrices in the test set.

Table 1. Parameters of MND for each class and transition matrix \mathcal{P} .

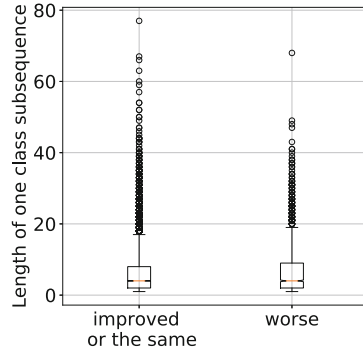
Class	M	U	V
1	$0.1J_8$	I_8	D_8
2	$0.2J_8$	D_8	I_8
3	$0.5J_8$	D_8	D_8

$$\mathcal{P} = \begin{bmatrix} 0.9 & 0.1 & 0 \\ 0.1 & 0.8 & 0.1 \\ 0 & 0.15 & 0.85 \end{bmatrix} \quad (14)$$

In order to analyze the impact of value of parameter Δ_p on the MCLMCF performance, we performed tests for $\Delta_p = [0.01, 0.02, 0.05, 0.1, 0.2, 0.5]$. Table 2, which presents the results obtained with the MCL and MCLMCF with different Δ_p parameter, reveals that increasing the value of Δ_p improves the classification performance of the MCLMCF. We can observe that the proposed classifier outperforms the MCL. We also analyzed the characteristics of sequences in the test set. We divided sequences into two groups: the first group (improved or the same) contains 411 sequences that were classified with higher or same accuracy with the use of MCLMCF with $\Delta_p = 0.5$, the second group (worse) comprises 75 sequences that were better classified with the MCL. Then we measured the

Table 2. Left panel – the results of the experiments for different values of Δ_p . Right panel – the lengths of homogeneous class sub-sequences in the test set.

Classifier	Mean acc	Var of acc
$\Delta_p = 0.5$	0.7757	0.0142
$\Delta_p = 0.2$	0.7627	0.0154
$\Delta_p = 0.01$	0.7237	0.0167
MCL	0.7212	0.0142
$\Delta_p = 0.1$	0.7190	0.0199
$\Delta_p = 0.05$	0.7165	0.0185
$\Delta_p = 0.02$	0.6913	0.0206



lengths of the homogeneous class subsequences (Table 2 - right panel). Our analysis shows that the proposed classifier performs better than MCL on sequences that contain long subsequences of the same class.

7 Case Study – Classifying an Image Series of Gas Burner Flames

The proposed approach was also tested on an image series of industrial gas burner flames. The contents of methane in natural gas fluctuate over longer periods of time, leading to proper, admissible or improper combustion that can be compensated by regulating the air rate supply. Thus, the contents of methane in the gas have all the features of the reoccurring concept drift. The mode of the combustion is easy to classify by an operator simply by visual inspection since the flame is either blue, blue and yellow, but still laminar or only yellow with a turbulent flow (see [19] and [20] for images and more details). It is however more difficult to derive an image classifier that works properly in heavy industrial conditions.

In order to test the Skeletal algorithm with the MC predictor, we selected the support vector machine (SVM) as the classifier in Step 1. The learning sequence consisted of 90 images in the series while the testing sequence length was 55. Having such a small testing sequence, we decided to extend it by adding 100 of its repetitions with the noise added (see [1] for the survey of methods of testing classifiers and to [11] for handling possibly missing data). The noise was rather heavy, namely, it was uniformly distributed in $[-0.25, 0.25]$ for grey level images represented in $[0, 1]$ interval.

In this study, we apply the estimated transition matrix shown in Fig. 2(left panel). Properties of the MC with matrix $\hat{\mathcal{P}}$ are listed in Table 3(right panel). Alternatively, one can consider the optimization of this matrix so as to attain better classification accuracy. This is however not such an easy task because we

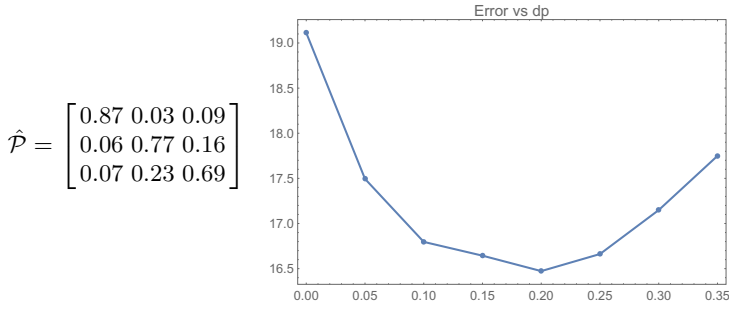


Fig. 2. Left panel – estimated \mathcal{P} matrix of the MC. Right panel – the classification error vs Δ_p .

Table 3. Left panel – the fraction of the mean occupation time (fMOT) for the labels of the learning system (2nd column) and for the states of MC \mathcal{P} (3rd column) Right panel – basic properties of MC \mathcal{P} .

State	fMOT L.S.	fMOT MC \mathcal{P}	
1	0.367	0.365	”Communicating”, [1, 2, 3], ”RecurrentClasses”, [1, 2, 3], ”Irreducible”, True, ”Aperiodic”, True
2	0.344	0.365	
3	0.289	0.270	

have to take into account many constraints (see [21] for a global search algorithm that is able to handle complicated constraints).

The mean occupation time (MOT) of the MC state is the number of discrete times (steps of the MC) that are spent by the MC in a given state. We apply this notion also to the labels of the learning sequence. By the fraction of the MOT (fMOT) we understand MOT’s of the states divided by the length of the observations or the length of the learning sequence, respectively. The comparison of the second and the third columns of Table 3 shows a very good agreement of the fMOT’s, which justifies the usage of our prediction model.

Then the Skeletal algorithm with $\gamma = 0.5$ and SVM as the basic classifier was run 100 times for Δ_p varying from 0 (i.e., there is no feedback from the MC predictor) to 0.35. As one can observe in Fig. 2 (right panel), for properly selected $\Delta_p = 0.2$ we can reduce the classification error by about 3 % in comparison to the case when the MC feedback is not applied. For long runs of industrial gas burners, the improvements of the decisions by 3 % may lead to essential savings. On the other hand, the same figure indicates the optimum is rather flat and selecting Δ_p in the range for 0.1 to 0.25 provides similar improvements.

8 Concluding Remarks

The problem of classifying a series of images with a reoccurring concept drift was considered. The class of algorithms for such problems was proposed and tested

both on simulated and industrial images. The idea of introducing feedback from the MC predictor of class labels to a priori class probabilities occurred to be fruitful in both cases, leading to the reduction of the classification error by about 3%, especially when we observe a long series of images from the same class.

Possible applications of the proposed approach include a long image series arising in the medical diagnostics (see, e.g., [5]). One can also combine the idea of the MC predictor with classifiers that are based on deep neural networks, e.g., in the version that was proposed in [6].

References

1. Anguita, D., Ghelardoni, L., Ghio, A., Ridella, S.: A survey of old and new results for the test error estimation of a classifier. *J. Artif. Intell. Soft Comput. Res.* **3**(4), 229–242 (2013)
2. Bersini, H., Saerens, M., Sotelino, L.G.: Hopfield net generation, encoding and classification of temporal trajectories. *IEEE Trans. Neural Netw.* **5**(6), 945–953 (1994)
3. Bifet, A.: Classifier concept drift detection and the illusion of progress. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2017. LNCS (LNAI)*, vol. 10246, pp. 715–725. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59060-8_64
4. Bilski, J., Smola, J.: Parallel realisation of the recurrent Elman neural network learning. In: Rutkowski, L., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2010. LNCS (LNAI)*, vol. 6114, pp. 19–25. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13232-2_3
5. Bruzdziński, T., Krzyżak, A., Fevens, T., Jeleń, L.: Web-based framework for breast cancer classification. *J. Artif. Intell. Soft Comput. Res.* **4**(2), 149–162 (2014)
6. Chang, O., Constante, P., Gordon, A., Singana, M.: A novel deep neural network that uses space-time features for tracking and recognizing a moving object. *J. Artif. Intell. Soft Comput. Res.* **7**(2), 125–136 (2017)
7. Devroye, L., Györfi, L., Lugosi, G.: *A Probabilistic Theory of Pattern Recognition*. Springer, Berlin (2013). <https://doi.org/10.1007/978-1-4612-0711-5>
8. Duda, P., Jaworski, M., Rutkowski, L.: On ensemble components selection in data streams scenario with reoccurring concept-drift. In: 2017 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–7. IEEE (2017)
9. Elman, J.L.: Finding structure in time. *Cogn. Sci.* **14**, 179–211 (1990)
10. Jaworski, M., Duda, P., Rutkowski, L.: Concept drift detection in streams of labelled data using the restricted Boltzmann machine. In: 2018 International Joint Conference on Neural Networks (IJCNN), pp. 1–7. IEEE(2018)
11. Jordanov, I., Petrov, N., Petrozziello, A.: Classifiers accuracy improvement based on missing data imputation. *J. Artif. Intell. Soft Comput. Res.* **8**(1), 31–48 (2018)
12. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: a survey. *Inf. Fusion* **37**, 132–156 (2017)
13. Kurzynski, M., Zolnierak, A.: A recursive classifying decision rule for second-order Markov chains. *Control Cybern.* **9**(3), 141–147 (1980)
14. Kurzynski, M., Majak, M.: Meta-Bayes classifier with Markov model applied to the control of bioprosthetic hand. In: Czarnowski, I., Caballero, A.M., Howlett, R.J., Jain, L.C. (eds.) *Intelligent Decision Technologies 2016. SIST*, vol. 57, pp. 107–117. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-39627-9_10

15. Krzyśko, M., Skorzybut, M.: Discriminant analysis of multivariate repeated measures data with a Kronecker product structured covariance matrices. *Stat. Pap.* **50**(4), 817–835 (2009)
16. Manceur, A.M., Dutilleul, P.: Maximum likelihood estimation for the tensor normal distribution: algorithm, minimum sample size, and empirical bias and dispersion. *J. Comput. Appl. Math.* **239**, 37–49 (2013)
17. Ohlson, M., Ahmad, M.R., Von Rosen, D.: The multilinear normal distribution: introduction and some basic properties. *J. Multivar. Anal.* **113**, 37–47 (2013)
18. Pajares, G.: A Hopfield neural network for image change detection. *IEEE Trans. Neural Netw.* **17**(5), 1250–1264 (2006)
19. Rafajłowicz, E., Pawlak-Kruczek, H., Rafajłowicz, W.: Statistical classifier with ordered decisions as an image based controller with application to gas burners. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2014. LNCS (LNAI)*, vol. 8467, pp. 586–597. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07173-2_50
20. Rafajłowicz, E., Rafajłowicz, W.: Image-driven decision making with application to control gas burners. In: Saeed, K., Homenda, W., Chaki, R. (eds.) *CISIM 2017. LNCS*, vol. 10244, pp. 436–446. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59105-6_37
21. Rafajłowicz, W.: Method of handling constraints in differential evolution using Fletcher’s filter. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *ICAISC 2013. LNCS (LNAI)*, vol. 7895, pp. 46–55. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38610-7_5
22. Rafajłowicz, E.: Data structures for pattern and image recognition with application to quality control. *Acta Polytechnica Hungarica Informatica* **15**(4), 233–262 (2018)
23. Rafajłowicz, E.: Classifiers for matrix normal images: derivation and testing. In: Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (eds.) *ICAISC 2018. LNCS (LNAI)*, vol. 10841, pp. 668–679. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91253-0_62
24. Rafajłowicz, E.: Robustness of raw images classifiers against the class imbalance – a case study. In: Saeed, K., Homenda, W. (eds.) *CISIM 2018. LNCS*, vol. 11127, pp. 154–165. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99954-8_14
25. Rafajłowicz, E.: Classifying image sequences with the Markov chain structure and matrix normal distributions. In: Rutkowski, L., Scherer, R., Korytkowski, M., Pedrycz, W., Tadeusiewicz, R., Zurada, J.M. (eds.) *ICAISC 2019. LNCS (LNAI)*, vol. 11508, pp. 595–607. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-20912-4_54
26. Skubalska-Rafajłowicz, E.: Sparse random projections of camera images for monitoring of a combustion process in a gas burner. In: Saeed, K., Homenda, W., Chaki, R. (eds.) *CISIM 2017. LNCS*, vol. 10244, pp. 447–456. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59105-6_38
27. Werner, K., Jansson, M., Stoica, P.: On estimation of covariance matrices with Kronecker product structure. *IEEE Trans. Signal Process.* **56**(2), 478–491 (2008)
28. Wozniak, M.: Markov chains pattern recognition approach applied to the medical diagnosis tasks. In: Oliveira, J.L., Maojo, V., Martín-Sánchez, F., Pereira, A.S. (eds.) *ISBMDA 2005. LNCS*, vol. 3745, pp. 231–241. Springer, Heidelberg (2005). https://doi.org/10.1007/11573067_24
29. Wójcik, W., Kotyra, A.: Combustion diagnosis by image processing. *Photonics Lett. Poland* **1**(1), 40–42 (2009)