Contents lists available at ScienceDirect

Artificial Intelligence in Medicine

journal homepage: www.elsevier.com/locate/aiim

Evolving fuzzy medical diagnosis of Pima Indians diabetes and of dermatological diseases

Stavros Lekkas [*], Ludmil Mikhailov

Decision Sciences Research Group, Manchester Business School East – F25, The University of Manchester, Manchester M15 9EP, United Kingdom

ARTICLE INFO

Article history:
Received 17 June 2008
Received in revised form 22 May 2010
Accepted 26 May 2010

Keywords:
Evolving fuzzy rule-based classification
Evolving medical diagnosis
Buffering of input samples
Interpretable medical rules

ABSTRACT

Objective: This paper reviews a methodology for evolving fuzzy classification which allows data to be processed in online mode by recursively modifying a fuzzy rule base on a per-sample basis from data streams. In addition, it shows how this methodology can be improved and applied to the field of diagnostics, for two popular medical problems.
Method: The vast majority of existing methodologies for fuzzy medical diagnostics require the data records to be processed in offline mode, as a batch. Unfortunately this allows only a snapshot of the actual domain to be analysed. Should new data records become available they require cost sensitive calculations due to the fact that re-learning is an iterative procedure. *eClass* is a relatively new architecture for evolving fuzzy rule-based systems, which overcomes these problems. However, it is data order dependent as different orders of the data result into different rule bases. Nonetheless, it is shown that models of *eClass* can be improved by arranging the order of the incoming data using a simple optimization strategy.
Results: In regards to the Pima Indians diabetes dataset, an accuracy of 79.37% was obtained, which is 0.84% lower than the highest in the literature. The proposed optimization strategy increased the accuracy and specificity of the model by 4.05% and 7.63% respectively. For the dermatology dataset, an accuracy of 97.55% was obtained, which is 1.65% lower than the highest in the literature. In this case, the proposed optimization strategy improved the accuracy of the model by 4.82%. The improved algorithm has been compared to other existing algorithms and seems to outperform the majority.
Conclusions: This paper has shown that *eClass* can effectively be applied to the classification of diabetes and dermatological diseases from discrete numerical samples. The results of using a novel optimization strategy indicate that the accuracy of *eClass* models can be further improved. Finally, the system can mine human readable rules which could enable medical experts to gain better understanding of a sample under analysis throughout the traditional diagnostic process.

© 2010 Elsevier B.V. All rights reserved.

1. Introduction

During the last few years observations show an escalating use of fuzzy set theory in the field of medical diagnostics. Steimann [1] observes and further analyses the rapid growth of relevant research in this field. Obviously, fuzzy set theory responds effectively to the non-statistical uncertainty, which is circumscribed in problems of the medical domain. Zadeh, the instigator and creator of fuzzy set theory, said back in 1969 that "*. . . fuzzy sets are very much likely to be applied in the field of medical diagnostics . . .*" [2]. What once seemed to be a "prophecy" has actually been the onset of a successful series of significant applications, which aim to contribute further to diagnostics in the domain of medicine.

Artificial Intelligence is a broad field which is comprised of a wide spectrum of methods for modeling solutions to problems. The main disadvantage of those which are not based on fuzzy sets is that they do not provide any explanation of how their inference result has been acquired. Fuzzy sets on the contrary, fill in this gap by supporting linguistic descriptions and expressions, as stated by Kuncheva and Steimann [3]. Fuzzy systems usually generate human interpretable rules which take the form of IF-THEN statements. Such rules which correspond to the knowledge granules of the diagnostic systems can be comprehended by human operators.

This study approaches medical diagnostics from the fuzzy classification perspective. A machine learning method is used in order to predict the class labels on the basis of discrete numerical features and to mine the knowledge (rule) base of the system.

[*] Corresponding author. Tel.: +44 0161 306 3361.
E-mail addresses: stavros.lekkas@postgrad.mbs.ac.uk, xfactor@linuxmail.org (S. Lekkas).

0933-3657/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2010.05.007

Existing fuzzy classification approaches, as in [4–8], generally follow the guideline of being offline. This means that they process data in batch and they are adjusted to act upon static data, all of which have to be available a priori; there is no evidence on whether other conditional instances which belong to the same data domain can be handled by the particular methods so that the respective model accuracies will remain unaffected. For this reason Kuncheva and Steimann [3], characterize classical, fixed rule-based systems as "inadequate".

Conversely, evolving fuzzy classification systems (EFCS), such as *eClass*, operate by self-developing their fuzzy rule base in one pass on a per-sample basis. Thus, their fuzzy rule base is not fixed as with the offline counterparts but instead it can adapt to the information brought by data samples which arrive from massive data streams sequentially. The adaptation process does not require re-learning, because it is based on recursive calculations. This opposes to the highly iterative nature of offline fuzzy systems, which involve additional computational burden and which cannot cope with large numbers of input samples. EFCSs also preserve the linguistic interpretability of the generated rules and they have found numerous applications, such as in autonomous robotic navigation by Zhou and Angelov [9] and in real-time landmark recognition by the same authors [10]. In medical context, they have been used for the classification of electroencephalogram (EEG) signals by Xydeas et al. [11], of breast cancer by Lekkas and Mikhailov [12] and they have shown great potential for further utilization in the field of medical diagnostics. Consequently, there is a need for such online systems which can handle massive input data (e.g. patient records with numerical attributes) by self-improving their knowledge base automatically.

An evolving fuzzy classification methodology for diagnostics of two well known medical problems is presented in this study. The first one regards Pima Indians diabetes (PID), which is a binary classification task. The second one is about the classification of six dermatological diseases (DERM) and can thus be approached as a 6-class classification task.

Diabetes mellitus is a complex metabolic disorder which is characterized by persistent hyperglycemia and which is resulting from defects in insulin secretion, insulin action or both [13]. There have been two types of diabetes identified, *Type-1* (insulin dependent) and *Type-2* (non-insulin dependent). The dataset which is analysed in this study consists of samples which originate from a population of Pima Indians. These samples refer to discrete Type-2 positive and negative instances. According to Knowler et al., the Pima Indians of Arizona, USA, have the highest reported incidence of diabetes in the world [14]. In addition, the predominant Type-2 diabetes of their population is said to have slow and gradual commencement. As a consequence, the traditional diagnostic method which is partially based on the plasma glucose test may be delayed by up to 10 years as Holt and Hanley mention [13].

The second medical problem, which is considered, is related to the diagnosis of six dermatological diseases. They are *psoriasis*, *seboreic dermatitis*, *lichen planus*, *pityriasis rosea*, *chronic dermatitis* and *pityriasis rubra pilaris*, all well known by dermatologists. They are considered as erythemato-squamous diseases and cause some visible histopathological disorders on patients' skin. The diagnosis of erythemato-squamous diseases is a real problem in dermatology. Most of them share the quantifiable features of erythema and scaling with small divergence and which can be obtained by extensive analysis of the patient's skin sample under a microscope by an expert.

This study focuses on three major points. The first one is to show that there is space for improving the accuracy of *eClass* and to present two new medical applications of the improved method. The second one is to compare the accuracies of *eClass* to the accuracies of the most successful fuzzy classification systems in the literature and to demonstrate its competency. Finally, the third one is to suggest that EFCSs, such as *eClass*, can be used by medical experts who need to validate the traditional diagnostic result or to gain a better understanding of the sample under examination in order to carry out the traditional diagnostic procedure.

Section 2 reviews the *eClass* architecture, which is suitable for evolving fuzzy classification and online decision making. The main methodology of learning the diagnostic rules of inference from numerical data samples is presented in this section. Section 3 presents specific details of the datasets such as the variables and the sample distribution in the classes of each domain. Moreover, it presents a novel method which optimizes the performance of the *eClass* algorithm, by modifying the order of the input samples. Section 4 presents the experimental results and performance comparisons of the proposed method. Some of the latter are discussed in Section 5 and their algorithmic complexities are considered. Finally in Section 6 the paper is concluded.

2. A methodology for evolving fuzzy rule-based classifiers

This section explains the methodology of the *eClass* algorithm, which is suitable for the design of evolving fuzzy rule-based systems. Section 2.1 describes the type of the fuzzy rules and the procedure of online enumeration of their fuzzy sets. Sections 2.2 and 2.3 describe the processes of *learning* the fuzzy rules from online data and Section 2.4 the *inference* (or diagnostic) process.

2.1. Fuzzy rule representation and recursively enumerable fuzzy sets

A Takagi-Sugeno-Kang (TSK) fuzzy rule of first order can be formulated as shown by Formula (1).

$$R^i : IF(x_1\ close\ to\ x_1^{i*})AND(x_2\ close\ to\ x_2^{i*})AND\ \ldots\ AND(x_n\ close\ to\ x_n^{i*})$$
$$THEN\ y_c^i = f^i \qquad (1)$$

In Formula (1) R^i is the ith fuzzy rule; in the antecedent part (if-part) of the rule, $x = [x_1, x_2, \ldots, x_n]^T$ is the n-dimensional feature vector. A feature vector contains discrete numerical observations which comprise of each individual data sample. $x^{i*} = [x_1^{i*}, x_2^{i*}, \ldots, x_n^{i*}]$ are the fuzzy sets of the ith fuzzy rule and x^{i^*} is the prototype of the ith rule. In the consequent part (then-part), $y^i = [y_1^i, y_2^i, \ldots, y_C^i]$ is the C-dimensional fuzzy output, more details of which can be found in Section 2.3; $i \in [1, N_c]$, $c \in [1, C]$, N_c is the number of rules per class c and C is the number of classes of the classification problem.

The degree of proximity of a sample x to the prototypes of a rule (*"close to"* in Formula (1)) is defined by the Gaussian membership function shown by Eq. (2).

$$\mu_j^i = e^{-(1/2)(d_j^i/\sigma_j^i)^2} \qquad (2)$$

In Eq. (2) μ_j^i is the membership of a sample x to the jth fuzzy set of the ith fuzzy rule, $j \in [1, n]$; d_j^i is the Euclidian distance of the sample x from the jth prototype of the ith rule; σ_j^i is the spread of the membership function, which defines the radius of influence of the jth prototype of the ith rule.

In earlier version of *eClass* [15] the parameter σ_j^i used to be a predefined constant which would be the same for all n dimensions. The main disadvantage of this choice is that the areas of projection of the fuzzy sets on the data space are separate subspaces which take the exact same shape. This reduces the flexibility of the model for it does not approximate efficiently the distribution of the data on the n-dimensional space.

In later versions [16,17] a more resourceful approximation method has been suggested to overcome this issue. In this case the areas of projection of the fuzzy sets are separate subspaces on the data space which take individual shapes. The reason is due to the fact that the spread of the data distribution can be recursively approximated based on existing information and on new one which is provided by new subsequent data samples.

The value of the spread of the very first rule, is initialized to 1, $\sigma^1 = 1$. For increasing values of the time step t, greater than 1, it can be obtained by Eq. (3).

$$\sigma_c^{t>1} = \rho\sigma_c^{t-1} + (1-\rho)\sqrt{(\sigma_c^{t-1})^2 + \frac{\left(\|x^{i*} - x^t\|^2 - (\sigma_c^{t-1})^2\right)}{S_c^t}} \tag{3}$$

In Eq. (3) σ_c^t is the current value of the spread; σ_c^{t-1} is the previous value of the spread; x^{i^*} is the prototype with the minimum Euclidian distance from the current input sample x^t; S_c^t is the number of previous input samples which have been associated to the prototype x^{i^*} (the support). Finally ρ, where $\rho \in [0,1]$, is a predefined constant, which regulates the approximation of the spread. Its value does not depend on the data domain. Angelov and Zhou [18] suggest the value of $\rho = 0.5$, which registers new information to be as much valuable as the existing one.

In the exceptional case that a brand new rule is generated, then its spread is initialized to the average value of the spreads of the existing N_c rules for that class, c, as shown by Eq. (4).

$$\sigma_t^{N_c+1} = \frac{\sum_{i=1}^{N_c} \sigma_t^i}{N_c} \tag{4}$$

2.2. Learning the antecedent part of the rules

The process of learning the antecedent (if-part) of the fuzzy rules of Eq. (1) is based on an online fuzzy clustering method called *eClustering* [16,17]. It extends Chiu's offline subtractive clustering method [19] with the intention to recursively partition online data, such as data streams, into clusters. It does not require the data samples to be labeled and thus it is as well suitable for unsupervised learning.

eClustering is prototype-based in the sense that the cluster centroids are actual data samples (not statistical means), which have been processed before the current one. In this context, a cluster can also refer to a fuzzy rule and the centroid of the cluster can refer to the prototype of that rule. The main idea is that each subsequent data sample is attributed to a *potential*, which describes its fitness to be the centroid of a cluster by approximating the density of the data. The potential of a sample to be a new cluster centroid is inversely proportional to the sum of the Euclidian distances between the current sample and all the previously processed ones. If the potential of a sample is higher than that of all the previous ones, then it retains strong candidacy to become a cluster centroid. It can recursively be approximated by using Eq. (5).

$$P_t(x^t) = \frac{S_c^t - 1}{a^t(S_c^t - 1) + \beta^t - 2\gamma^t + (t-1)}; t = 2, 3, \ldots \tag{5}$$

In Eq. (5), x^t is the current sample, the potential of which needs to be identified; S_c^t is the current total support of clusters of class c; t is the time step identifier. In the case that $t = 1$ the sample x^1 initializes the fuzzy rule base and its potential is set to 1, $P_1(x^1) = P_1(x^{1*}) = 1$. It should be noted that the number of classes C does not need to be set a priori. Every sample x^t which belongs to a new class can be used to initialize a new class-representative rule in online mode, for $t > 1$. Finally the variables α, β and γ are used as accumulators for the recursive calculation of the potential, which otherwise would require explicit storage of all the data history. They can be updated as shown by Eqs. (6a)–(6d).

$$\alpha^t = \sum_{j=1}^{n} (x_j^t)^2 \tag{6a}$$

$$\beta^t = \beta^{t-1} + \alpha^{t-1}; \quad \beta^1 = 0 \tag{6b}$$

$$\gamma^t = \sum_{j=1}^{n} x_j^t \Gamma_j^t \tag{6c}$$

$$\Gamma_j^t = \Gamma_j^{t-1} + x_j^{t-1}; \quad \Gamma_j^1 = 0 \tag{6d}$$

The process of calculating the potential of a sample should always be followed by revision of the potential values of the existing fuzzy rule prototypes. The reason is that the new sample brings new information, which has to be integrated to the existing model by reassessing the existing potential values. This calculation can be made recursively as shown by Eq. (7).

$$P_t(x_t^{i*}) = \frac{(S_c^t - 1)P_{t-1}(x_t^{i*})}{S_c^t - 2 + P_{t-1}(x_t^{i*}) + P_{t-1}(x_t^{i*})\sum_{j=1}^{n} \|x_t^{i*} - x_t\|_j^2}; \quad t = 2, 3, \ldots \tag{7}$$

In Eq. (7) S_c^t is the total support of clusters of class c; x_t^{i*} is the prototype subject to recalculation; x_t is the current sample, the new piece of information to be considered.

The structure of the fuzzy rule base is not fixed, but evolves according to two main criteria. The first one as shown by Formula (8) determines the formation of a new rule and causes long term expansion of the rule base. The Boolean condition A of Formula (8) can be true only when the potential of the current sample x^t is greater than the maximum potential of all N prototypes x^{i^*}.

$$\boldsymbol{A}: \quad P_t(x^t) > \max_{i=1}^{N} P_t(x_t^{i*}) \tag{8}$$

$$\boldsymbol{B}: \quad \boldsymbol{A}\, AND\, (\exists\, i, i \in [1, N]; \mu_j^i(x_j^t) > \kappa\, \forall\, j, \quad j \in [1, n]) \tag{9}$$

The second criterion as shown by Formula (9) regards the modification of existing fuzzy rules and guarantees long term shrinking (simplification) of the fuzzy rule base. The second conjunctive Boolean sub-condition of B in Formula (9) is only satisfied when the current data sample is too close to the nearest prototype. The degree of closeness of the two samples can be expressed in terms of the degree of membership μ as described in Section 2.1 and shown by Eq. (2). Thus, instead of generating a new rule which would be rather similar to an existing one, the centroid of nearest existing rule is just replaced by the current sample x_t. In Formula (9), κ is a threshold. Meaningful values of κ are usually in the range $[0.1, e^{-1}]$. The value of $\kappa_{DERM} = \kappa_{PID} = e^{-1}$ is suggested for the analysis of the DERM and PID datasets respectively.

This section has described the *eClustering* method. It includes a data density measure and two primary conditions in order to control the structural changes of a fuzzy rule base. Learning the antecedent part of the fuzzy diagnostic rules is made on a per-sample basis using recursive calculations.

2.3. Learning the consequent part of the rules

The process of learning the consequent of a first order TSK fuzzy rule (refer to Formula (1)) is realized by approximating the non-linear n-dimensional input using multiple locally linear sub-models, one per class c of the C-class problem. In this case a fuzzy rule can have C consequents, each of which estimates the possibility of an input sample to belong to a certain class *c*. An extended form of the fuzzy rule consequent of Formula (1) is given by Eq. (10).

$$f^i = [1, x_1, x_2, \ldots, x_n] \begin{pmatrix} \theta_{01}^i & \cdots & \theta_{0C}^i \\ \vdots & \ddots & \vdots \\ \theta_{n1}^i & \cdots & \theta_{nC}^i \end{pmatrix} \tag{10}$$

In Eq. (10), n is the number of dimensions and the vector $\bar{x} = [1, x_1, x_2, \ldots, x_n] = [1, x^T]$ is the extended input data surface; θ_{jc}^i is the jth linear parameter (invariant) of the cth sub-model of the ith rule, $j \in [1, n]$, $c \in [1, C]$, $i \in [1, N]$. Let Θ be the matrix containing all the linear parameters. This matrix requires $(n + 1)C$ space capacity. The identification of the linear parameters is based on a fuzzy weighted Recursive Least Squares (wRLS) estimation, as suggested in [16,18] and further applied in [12]. The wRLS method is utilized as a means of minimizing the error between the input and output space and converges within every recursive step. The matrix Θ can be recursively identified and initialized per rule i as shown by Eq. (11).

$$\Theta_t^i = \Theta_{t-1}^i + C_t^i \lambda^i \bar{x}_t (\bar{y}_t - x_t^T \Theta_{t-1}^i); \quad \Theta_1^i = 0 \tag{11}$$

$$C_t^i = C_{t-1}^i - \frac{\lambda^i C_{t-1}^i \bar{x}_t x_t^T C_{t-1}^i}{1 + \lambda^i x_t^T C_{t-1}^i \bar{x}_t}; \quad C_1^i = \Omega I; \quad t = 2, 3, \ldots \tag{12}$$

$$\lambda^i(x^T) = \frac{\prod_{j=1}^{n} \mu_j^i(x_j^T)}{\sum_{k=1}^{N} \prod_{j=1}^{n} \mu_j^i(x_j^T)} \tag{13}$$

In Eq. (11), $\bar{y}_t$ is the genuine class label of the input sample x_t^T, in binary format; C^i is the covariance matrix of the ith rule, which can also be recursively updated and initialized as shown by Eq. (12). In the initialization phase, of Eq. (12), Ω is a large constant and I is the identity matrix. For the experimental results of this study, it has been assumed that $\Omega = 50$. The size of each covariance matrix is $(n + 1)(n + 1)$, thus the algorithmic complexity to perform N RLS estimations (one per rule) is a square of the number of dimensions, $O(N(n + 1)^2)$.

In both Eqs. (11) and (12), λ^i is the normalized firing rate of the ith rule. This variable indicates the strength of the rule and is used as a weight in each of the RLS procedures. It can be obtained as shown by Eq. (13). Once the matrix with the linear parameters has been identified, the cth possibility of a sample x^T can be obtained by Eq. (14). In Eq. (14), $i \in [1, N]$, $c \in [1, C]$ and $j \in [1, n]$.

$$y_c^i = \lambda^i \left(\theta_{0c}^i + \sum_{j=1}^{n} \theta_{jc}^i x_j^T \right) \tag{14}$$

This section has described the process of learning the consequent of first order TSK fuzzy rules. In the consequent of such a rule, the number of outputs is equal to the number of classes and each output is an estimation of the possibility of an input sample to belong to a particular class. However, the wRLS method which aims to approximate these possibilities requires the real label of the sample. Therefore, learning the consequent part of a fuzzy rule is a supervised procedure.

2.4. The rule inference process

The inference process complements the one of learning, since without the former the latter is inessential. The system is aimed to learn in order to make predictions, which are based on the product of its learning, its knowledge. And it would be even more useful if it could display its rules of inference in a human readable form.

The ith TSK fuzzy rule of first order, as shown by Formula (1), includes a consequent part y^i which is comprised by C different fuzzy outputs, $y^i = [y_1^i, y_2^i, \ldots, y_C^i]$. Section 2.3 has described that each of these outputs specifies the possibility of a sample to belong to the cth class of a C-class problem and suggests a method to work them out. Though, a rule cannot be read and comprehended by human operators in its fuzzy form, unless it gets defuzzified.

The defuzzification of a fuzzy rule is a process of transforming it into a crisp one. Kuncheva and Steimann [3], consider the paradoxical need for this operation by posing the following (rhetorical) question: "*If we really need crisp answers, then why bother using fuzzy sets in the first place?*". Although an *eClass* model makes sole use of its knowledge in its fuzzy form, a defuzzification process is also employed for reasons of transparency. Specifically, the C-fold fuzzy output y^i is replaced by a label which indicates a particular diagnostic result. Using the *max* operator this replacement can be made in the consequent of the ith rule as shown by Eq. (15). In Eq. (15), $Class^i$ is the class label of the ith rule, which corresponds to the class c of the problem with the highest possibility y_c^i as by Eq. (14).

$$Class^i = \operatorname*{argmax}_{c=1}^{C} (y_c^i) \tag{15}$$

3. On the data

This section provides description of the variables, classes and sample class distributions of the PID and DERM datasets. In addition it explains a common characteristic of online classification systems and based on that it presents a novel method, which can improve the accuracy of *eClass*.

3.1. The datasets

The PID and DERM datasets, which are examined in this study, are available from the UCI Machine Learning Repository at [20,21] respectively.

The PID dataset contains 768 data samples and 8 numerical features per sample (sample dimensionality). The variables are allocated as shown in Table 1. Each of the samples contains a label which indicates the class of the sample. There are two classes in total. The first class is labeled as "*negative to diabetes*" and involves 500 samples (65.10% of the dataset) whilst the second one is labeled as "*positive to diabetes*" and involves 268 samples (34.89% of the dataset). The fact that it is a binary classification problem enables the use of the sensitivity and specificity quality measures, in addition to the accuracy.

The DERM dataset contains 366 data samples, 8 of which have been removed for they contain missing values. Each sample

Table 1
The variables of the PID dataset.

ID	Attribute	ID	Attribute
1	No. of times pregnant	5	2-h serum insulin (μU/ml)
2	Plasma glucose concentration	6	Body mass index (kg/m^2)
3	Diastolic blood pressure (mm Hg)	7	Diabetes pedigree function
4	Triceps skin fold thickness (mm)	8	Years of age

Table 2
The clinical features of the DERM dataset.

ID	Attribute	ID	Attribute
1	Erythema	7	Follicular papules
2	Scaling	8	Oral mucosal involvement
3	Definite borders	9	Knee and elbow involvement
4	Itching	10	Scalp involvement
5	Koebner phenomenon	11	Family history (0 or 1)
6	Polygonal papules	34	Age (linear)

Table 3
The histopathological features of the DERM dataset.

ID	Attribute	ID	Attribute
12	Melanin incontinence	23	Spongiform pustule
13	Eosinophils in the infiltrate	24	Munro microabcess
14	PNL infiltrate	25	Focal hypergranulosis
15	Fibrosis of the papillary dermis	26	Disappearance of the granular layer
16	Exocytosis	27	Vacuolisation and damage of basal layer
17	Acanthosis	28	Spongiosis
18	Hyperkeratosis	29	Saw-tooth appearance of retes
19	Parakeratosis	30	Follicular horn plug
20	Clubbing of the rete ridges	31	Perifollicular parakeratosis
21	Elongation of the rete ridges	32	Inflammatory monoluclear inflitrate
22	Thinning of the suprapapillary epidermis	33	Band-like infiltrate

Table 4
The class distribution of the DERM dataset.

Dermatology dataset information		
Class label	Class distribution	No of samples
Psoriasis	30.73%	110
Seboreic dermatitis	17.04%	61
Lichen planus	19.83%	71
Pityriasis rosea	13.41%	48
Chronic dermatitis	13.41%	48
Pityriasis rubra pilaris	5.58%	20
Total	100.00%	358

consists of 34 numerical features, 12 of which are of clinical nature and 22 of histopathological nature. The description of these features is given by Tables 2 and 3. The dimensionality of this dataset is large enough to stress the *eClass* algorithm and to determine its effectiveness in real-time processing. The samples belong to one of the six classes of the problem. The class labels and sample distributions are given by Table 4.

Finally it should be noted that the *eClass* methodology is intended for online data from data streams and not from datasets. Thus, it is assumed that the data samples in these datasets are discrete instances of a data stream and they are loaded and processed on a per-sample basis.

3.2. The data order factor

According to Cornuejols [22], order independent incremental systems have the ability to memorize each individual sample processed in the past. Thus they are iterative and require le-learning from the complete data history, which consequently demands explicit storage. However, the order by which the data samples are loaded from a data stream and processed by *eClass* plays a key role to the accuracy of the model. This sort of online incremental systems is data order dependent because they store representative prototypes (refer to Section 2.2) only, which are only part of the complete data history. To put it in simple words, the different order of the samples results to extraction of dissimilar knowledge structures and thus to divergent accuracies for the same models.

From experiments conducted on the *eClass* algorithm it has been observed that the average model accuracy increases for specific data order patterns. More specifically, if the samples are reorganized into chunks which can be indicated by their unique class label, then the model accuracy will gradually increase. Based on this observation a *buffering* technique is proposed, which assumes that the samples are labeled. It is similar to the one described by Aggarwal et al. [23], in the sense that the samples are processed in time windows and not in batch.

By definition, systems such as *eClass* cannot explicitly "look back", but only in terms of stored landmarks (the cluster prototypes). However, the fact that the samples can be provisionally stored before passed to the *eClass* algorithm enables the system to "look ahead". The proposed technique endeavors to rearrange the order of the data samples in small time windows by storing each one into a specific buffer. The number of buffers should be equal to the number of classes C of the classification problem. All these buffers are of fixed, equal size s and thus the algorithmic complexity is just $O(C \times s)$. A simple example of this concept is illustrated by Fig. 1.

From the example of Fig. 1 one can observe that there are 3 classes for this fictional problem and thus 3 buffers have been justified. Each subsequent sample of the data stream is temporarily stored into a buffer, purposely allocated to capacitate 5 samples of its class. When a buffer has no more free space then it passes all its 5 contents to the *eClass* algorithm in strict first-in-first-out (FIFO) order and its space is again free to store other forthcoming samples and so on. The data order which is imposed by the application of the proposed buffering method is no longer matching the original one.

It is imperative to clarify two details in regards to the use of the proposed technique. The first one is that the number of buffers does not need to be predetermined as the insertion of a sample of a new class can trigger the instantiation of a new buffer and the immediate storage. Therefore the peak complexity is only reached when samples of all classes have been imported, with all the buffers being full to the limit. The second detail regards the type of the input source. Assuming that data streams are never-ending input sources, we can conclude to that there will not be any remainders in the buffers, which have not been sent to the *eClass* algorithm. However this is not the case with sources of finite input, such as with datasets. With datasets, some buffers might not get filled to their top from the last few samples of the dataset and thus these samples will never be exported to the algorithm. In this case, each buffer is forced to export its content in FIFO order until there are no samples remaining into the buffers.

4. Experimental results

This section presents the experimental results of applying the *eClass* algorithm on the PID and DERM datasets. Four different

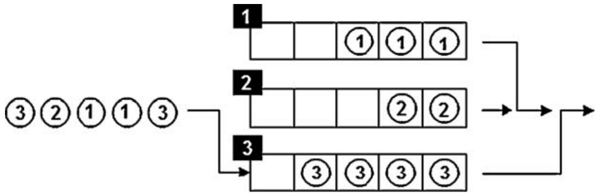


Fig. 1. A simple example of the buffering strategy applied to online data.

Table 5
Performance of *eClass* for different orders of the PID data.

Config.	Data order	Accuracy	Sensitivity	Specificity	No of rules	Execution time
#1	Original order	75.32%	69.50%	77.38%	12	296 ms
#2	2 buffers, capacity = 2 × 10	76.11%	65.44%	81.98%	15	421 ms
#3	2 buffers, capacity = 2 × 20	77.41%	67.53%	82.73%	10	327 ms
#4	2 buffers, capacity = 2 × 30	**79.37%**	**69.53%**	**85.01%**	**7**	265 ms

Table 6
Performance of *eClass* for random orders of the PID data.

Config.	Data order	Accuracy	Sensitivity	Specificity	No. of rules
#1	Random order 1	75.32%	68.22%	78.08%	13
#2	2 buffers, capacity = 2 × 10	77.93%	**75.78%**	78.64%	10
#3	2 buffers, capacity = 2 × 20	77.54%	72.51%	79.45%	15
#4	2 buffers, capacity = 2 × 30	**78.06%**	73.46%	**79.82%**	20
#1	Random order 2	76.11%	71.00%	77.91%	14
#2	2 buffers, capacity = 2 × 10	77.41%	72.59%	79.21%	11
#3	2 buffers, capacity = 2 × 20	77.54%	72.72%	79.35%	13
#4	2 buffers, capacity = 2 × 30	**78.21%**	**74.76%**	**80.21%**	18
#1	Random order 3	75.97%	67.81%	**79.55%**	10
#2	2 buffers, capacity = 2 × 10	75.06%	66.23%	78.94%	14
#3	2 buffers, capacity = 2 × 20	75.71%	67.08%	79.58%	15
#4	2 buffers, capacity = 2 × 30	**76.24%**	**69.05%**	79.19%	11
#1	Random order 4	76.11%	71.42%	77.71%	14
#2	2 buffers, capacity = 2 × 10	77.54%	74.86%	78.43%	8
#3	2 buffers, capacity = 2 × 20	77.41%	74.47%	78.39%	12
#4	2 buffers, capacity = 2 × 30	**77.80%**	**75.39%**	**78.60%**	12
#1	Random order 5	75.45%	69.08%	77.81%	7
#2	2 buffers, capacity = 2 × 10	75.45%	67.87%	78.53%	13
#3	2 buffers, capacity = 2 × 20	77.41%	72.59%	79.21%	22
#4	2 buffers, capacity = 2 × 30	**78.32%**	**74.63%**	**79.67%**	11
#1	Random order 6	76.76%	69.95%	79.55%	13
#2	2 buffers, capacity = 2 × 10	77.15%	70.53%	79.88%	9
#3	2 buffers, capacity = 2 × 20	75.97%	68.77%	78.89%	8
#4	2 buffers, capacity = 2 × 30	**77.80%**	70.63%	**80.97%**	11

configurations (#1–4) have been used per dataset and the respective performances have been subject of comparison. The first configuration indicates that the data will be processed in their original order. The rest three configurations utilize the buffering technique, which has been proposed in Section 3.2 in order to realize different data orders. Buffers that can store 10, 20 and 30 samples have been used respectively. In addition, some additional configurations have been included, which refer to 6 random orders of the original data, again with and without the utilization of buffers. Their aim is to justify the use of the buffering technique in conjunction with *eClass* on a wider application scale.

It should be noted that there is no need to separate any training and testing data partitions. The models predict the class label, update their structure and then use the real class label to identify

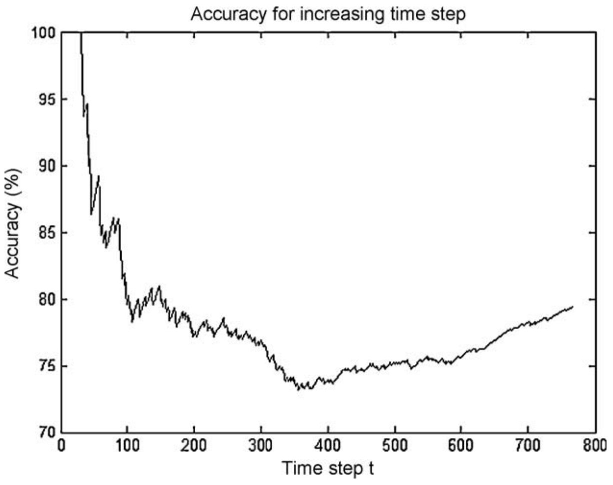


Fig. 2. The learning rate of *eClass & buffering* (*size* = *30*) on the PID dataset.

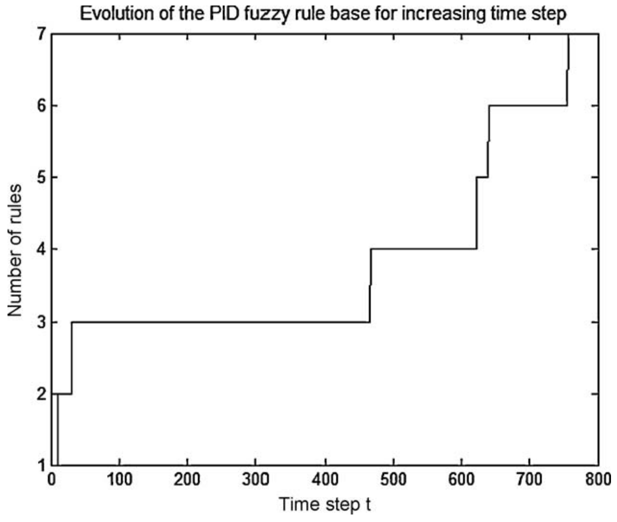
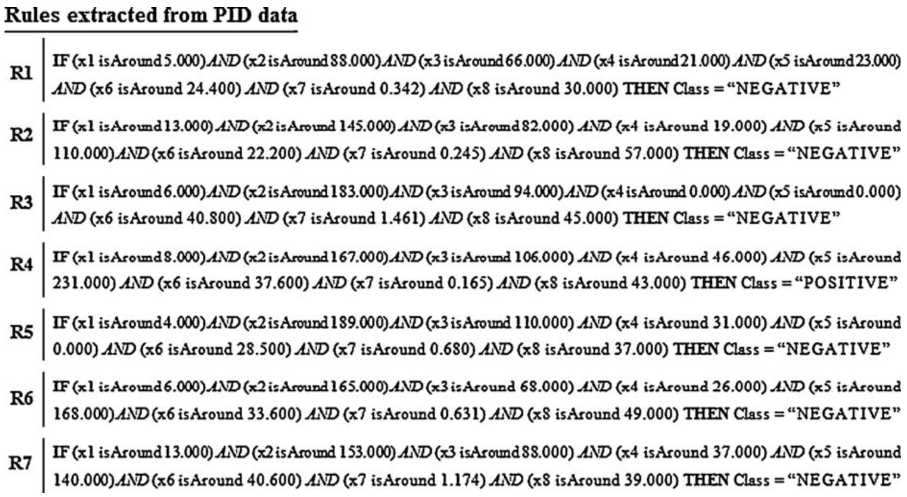


Fig. 3. Expansion of the rule base for increasing time step on the PID dataset.



Rules extracted from PID data

R1	IF (x1 isAround 5.000) *AND* (x2 isAround 88.000) *AND* (x3 isAround 66.000) *AND* (x4 isAround 21.000) *AND* (x5 isAround 23.000) *AND* (x6 isAround 24.400) *AND* (x7 isAround 0.342) *AND* (x8 isAround 30.000) THEN Class = “NEGATIVE”
R2	IF (x1 isAround 13.000) *AND* (x2 isAround 145.000) *AND* (x3 isAround 82.000) *AND* (x4 isAround 19.000) *AND* (x5 isAround 110.000) *AND* (x6 isAround 22.200) *AND* (x7 isAround 0.245) *AND* (x8 isAround 57.000) THEN Class = “NEGATIVE”
R3	IF (x1 isAround 6.000) *AND* (x2 isAround 183.000) *AND* (x3 isAround 94.000) *AND* (x4 isAround 0.000) *AND* (x5 isAround 0.000) *AND* (x6 isAround 40.800) *AND* (x7 isAround 1.461) *AND* (x8 isAround 45.000) THEN Class = “NEGATIVE”
R4	IF (x1 isAround 8.000) *AND* (x2 isAround 167.000) *AND* (x3 isAround 106.000) *AND* (x4 isAround 46.000) *AND* (x5 isAround 231.000) *AND* (x6 isAround 37.600) *AND* (x7 isAround 0.165) *AND* (x8 isAround 43.000) THEN Class = “POSITIVE”
R5	IF (x1 isAround 4.000) *AND* (x2 isAround 189.000) *AND* (x3 isAround 110.000) *AND* (x4 isAround 31.000) *AND* (x5 isAround 0.000) *AND* (x6 isAround 28.500) *AND* (x7 isAround 0.680) *AND* (x8 isAround 37.000) THEN Class = “NEGATIVE”
R6	IF (x1 isAround 6.000) *AND* (x2 isAround 165.000) *AND* (x3 isAround 68.000) *AND* (x4 isAround 26.000) *AND* (x5 isAround 168.000) *AND* (x6 isAround 33.600) *AND* (x7 isAround 0.631) *AND* (x8 isAround 49.000) THEN Class = “NEGATIVE”
R7	IF (x1 isAround 13.000) *AND* (x2 isAround 153.000) *AND* (x3 isAround 88.000) *AND* (x4 isAround 37.000) *AND* (x5 isAround 140.000) *AND* (x6 isAround 40.600) *AND* (x7 isAround 1.174) *AND* (x8 isAround 39.000) THEN Class = “NEGATIVE”

Fig. 4. The linguistically interpretable rule base of model #4 from the PID dataset.

the linear parameters on a per-sample basis. Thus, such models are subsequently tested on and trained by the same sample, for almost each sample of the dataset. However, in the very specific case that a sample belongs to a new class which has not been learned before, only the learning mechanism is invoked without the requirement to return a decision for that sample.

Table 5 presents the results of the analysis on the PID dataset. The proposed buffering technique has effectively improved the accuracy of *eClass*. All buffer-based models have outperform the first one, which has processed data according to their original order and model #4, which can store 30 samples per class, has given the best performance. The gain in performance from the latter is 4.05%, 0.03% and 7.63% in terms of accuracy, sensitivity and specificity respectively. Table 6 also justifies this increase in performance, in terms of 6 random orders of the PID data which differ from the original one.

Fig. 2 presents the graph of the learning rate of model #4, as by Table 5. From the graph one can observe a continuous loss of accuracy which takes place when the system processes samples of a different class and which is due to the expansion of the output space. However, the rate stabilized when half of the dataset had been processed. In addition, 7 interpretable rules have been extracted in total, which are by 5 rules less than those of the first configuration. The graph of rule base expansion for model #4 is given by Fig. 3. In this graph, the corners represent the creation of a new rule, whilst any rule modification is positioned somewhere in the flat horizontal regions. Fig. 4 shows the 7 human comprehensible rules which have been extracted by model #4. In this figure, the variables *x*1–*x*8 correspond to the 8 features which are shown by Table 1. The representation of the diagnostic rules is similar to the one suggested by Mencar et al. [31] but the respective consequents pole apart.

The execution time[1] of model #4, as by Table 5, has lasted 265 ms which gives an average processing time of 0.34 ms per data sample. This further signifies a computational power of 2941 data samples for every second of execution. Fig. 5 presents an approximation of the computational time of model #4 as a function of the time step *t*. Finally, the comparison provided in Table 7 shows that the enhanced model #4 has been more accurate than existing online ones and ranks second after an offline *General Regression Neural Network* (GRNN) [26].

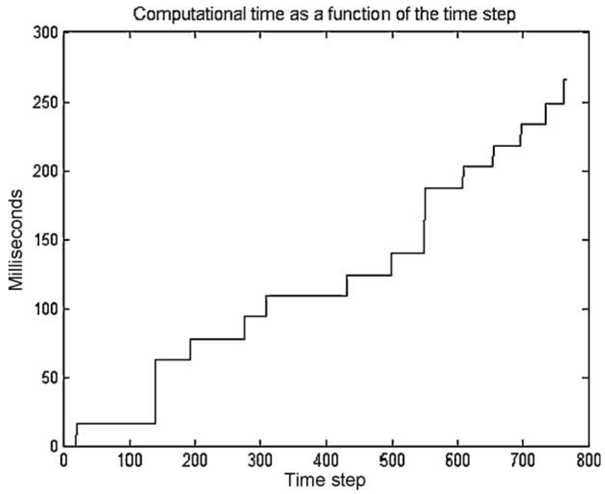


Fig. 5. The computational time of *eClass* as function of the time step on the PID dataset.

The results of applying the four aforementioned configurations on the DERM dataset are presented by Table 8. The buffering technique again seems to have a positive impact on the performance of the models #2–4, compared to the plain model #1. More specifically model #3, which uses buffers of 20 samples capacity, has reported a gain of 4.82% in accuracy but also an increase of the rule base by 4 rules. Figs. 6 and 7 present the graph of the learning rate and the expansion of the rule base for this model. Additionally, Table 9 shows a wider reflection of the performance amplification on the DERM data, which is due to the use of buffers.

Table 7
Comparative results with other methods on the PID dataset.

Method	Accuracy	Type
Fuzzy similarity classifier [24]	75.29%	Offline
Inverted Hierarchical Neuro-Fuzzy System [4]	78.60%	Offline
AWAIS [25]	75.87%	Offline
General Regression Neural Network [26]	**80.21%**	Offline
GDA-LSSVM [27]	79.16%	Offline
Incremental Random Forests [28]	76.80%	Online
Incremental SVM-FP [29]	73.80%	Online
Incremental PCA [30]	68.10%	Online
eClass, buffer capacity = 30 (this study)	**79.37%**	Online

[1] The experiments have been carried out on an Intel Pentium M 1.86 GHz with 1.5 GB of RAM.

Table 8
Performance of *eClass* for different orders of the DERM data.

Config.	Data order	Accuracy	No. of rules	Execution time
#1	Original order	92.73%	**19**	5228 ms
#2	6 buffers, capacity = 6 × 10	94.56%	21	5119 ms
#3	6 buffers, capacity = 6 × 20	**97.55%**	23	6688 ms
#4	6 buffers, capacity = 6 × 30	96.19%	23	6104 ms

The execution time of model #3, as by Table 8, has lasted 6688 ms. The difference between the execution times on the PID and DERM datasets can be justified from the fact that the latter has higher dimensionality (34 dimensions) than the former (8 dimensions). However, model #3 is still able to process each sample in 18.27 ms which signifies the speed of 54.73 samples for every second of execution. Fig. 8 presents an approximation of the computational time of model #3 as a function of the time step *t*. Finally, the comparative analysis of Table 10 shows that the enhanced *eClass* model #3 outperforms all existing online systems. Even so, it ranks second after an offline *Voting Feature Intervals* (VFI5) method [8].

5. Related works and discussion

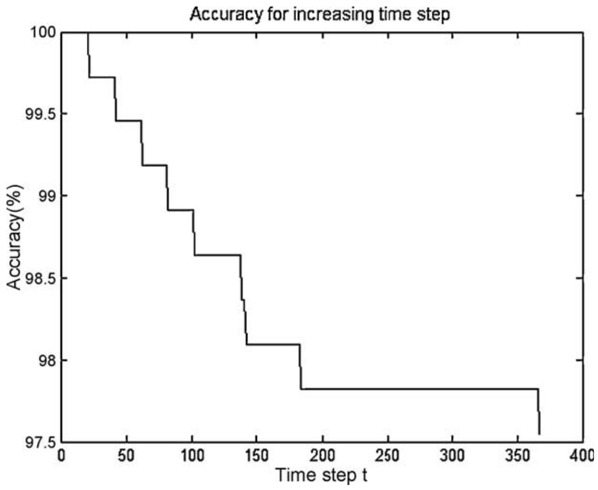


Fig. 6. The learning rate of *eClass & buffering* (*size* = *20*) on the DERM dataset.

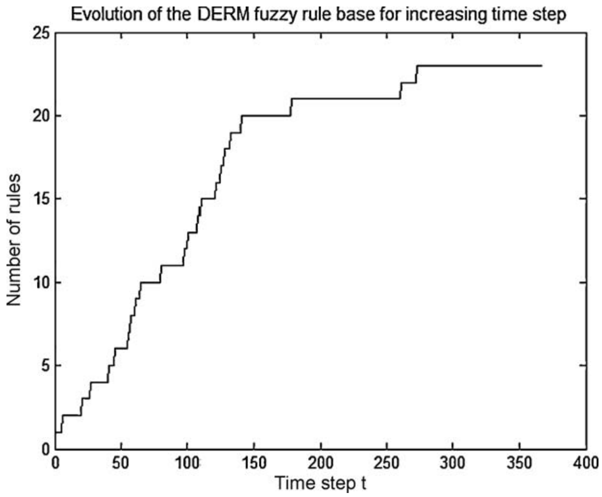


Fig. 7. Expansion of the rule base for increasing time step on the DERM dataset.

Table 9
Performance of *eClass* for random orders of the DERM data.

Config.	Data order	Accuracy	No of rules
#1	Random order 1	93.57%	22
#2	6 buffers, capacity = 6 × 10	94.97%	26
#3	6 buffers, capacity = 6 × 20	94.97%	25
#4	6 buffers, capacity = 6 × 30	**95.81%**	25
#1	Random order 2	92.17%	23
#2	6 buffers, capacity = 6 × 10	94.69%	20
#3	6 buffers, capacity = 6 × 20	95.53%	21
#4	6 buffers, capacity = 6 × 30	**97.20%**	18
#1	Random order 3	93.85%	19
#2	6 buffers, capacity = 6 × 10	94.41%	20
#3	6 buffers, capacity = 6 × 20	94.41%	27
#4	6 buffers, capacity = 6 × 30	**96.64%**	25
#1	Random order 4	93.29%	31
#2	6 buffers, capacity = 6 × 10	92.45%	26
#3	6 buffers, capacity = 6 × 20	93.01%	23
#4	6 buffers, capacity = 6 × 30	**96.08%**	27
#1	Random order 5	94.41%	21
#2	6 buffers, capacity = 6 × 10	95.53%	28
#3	6 buffers, capacity = 6 × 20	94.97%	23
#4	6 buffers, capacity = 6 × 30	**95.53%**	24
#1	Random order 6	92.73%	23
#2	6 buffers, capacity = 6 × 10	92.45%	24
#3	6 buffers, capacity = 6 × 20	94.69%	26
#4	6 buffers, capacity = 6 × 30	**95.81%**	24

Some of the algorithms which have been used for the comparisons in Section 4 are briefly reviewed in this section. Kayaer and Yildirim [26] have benchmarked three different neural network structures. A multilayer perceptron (MLP), a radial basis function (RBF) and a GRNN, have been trained and tested separately on the PID dataset. The findings indicate that although the MLP outperformed during the training phase, the GRNN was tested to be as much as 80.21% accurate.

Goncalves et al. [4] introduced an *Inverted Hierarchical Neuro-Fuzzy* system which has also been benchmarked on the PID dataset. The specific advantage of this method is that it employs a variable selection mechanism which arranges the variables in ascending order according to their training error. The testing accuracy of 78.60% was achieved.

In [8], Guvenir et al. propose the VFI5 algorithm. During its training phase, a VFI5 model learns how to differentiate concepts in the domain; the concepts are represented by intervals on each variable. During the classification, a real-valued voting process takes place and the class having collected the vote majority

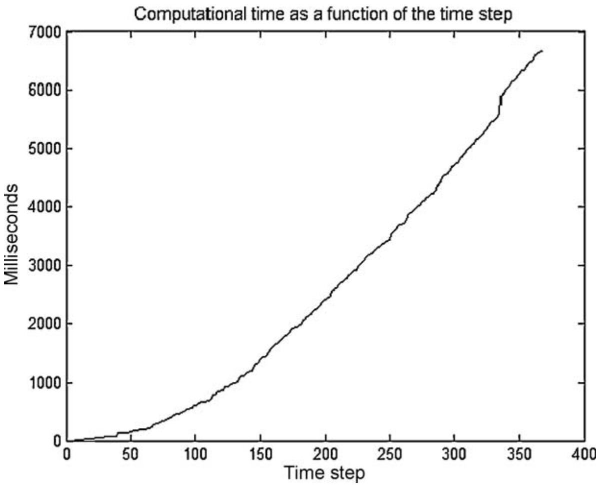


Fig. 8. The computational time of *eClass* as function of the time step on the DERM dataset.

Table 10
Comparative results with other methods on the DERM dataset.

Method	Accuracy	Type
Fuzzy ARTMAP [5]	93.14%	Offline
VFI5 [8]	**99.20%**	Offline
Fuzzy similarity classifier [24]	96.04%	Offline
Fast Global *k*-means [32]	93.51%	Online
Evolving NNTrees – Method 1 [33]	93.39%	Online
ESNTrees [34]	97.46%	Online
eClass, buffer capacity = 20 (this study)	**97.55%**	Online

becomes the predicted class. This model is as much as 99.20% accurate on the differential diagnosis of the 6 erythemato-squamous diseases, from the DERM dataset.

All the aforementioned algorithms are designed for offline use and hence they cannot manage online data efficiently from a computational point of view. That is mainly because they have to store all the data history and because they must bear re-training for every incoming sample. For example, most trained neural network models have exponential complexity (e.g.: O(2^n)) to the number of dimensions n of a sample [33], which is further increased by the number of re-training processes. Therefore, they might be well accurate but they fail due to the computational burden which results from the high-speed nature of data streams.

Park and Choi [30] have implemented a single-pass *incremental Principal Component Analysis* (iPCA) algorithm which has been applied on the PID dataset. This algorithm uses *Gram-Schmidt Orthogonalization* (GSO) in order to enforce the orthogonality of the eigenvectors, which improves the accuracy. In addition, it is covariance-free. Thus, it can identify eigenvectors faster than other covariance-based iPCA implementations. Its complexity is O(k^2n), where k is the number of eigenvectors and n the dimensionality. Comparatively, the *eClass* algorithm has square complexity too (refer to Section 2.3). Nevertheless, in conjunction with the buffering technique, it can be by as much as 11.27% more accurate on the PID domain as shown by Table 7.

6. Conclusions

This paper has presented a study of semi-supervised evolving fuzzy classification on the diagnostics for two medical problems. The two respective domains contain real records of patients with a known diagnosis. The objective is to review the existing methodology for evolving fuzzy classification, to improve it and finally to evaluate its performance in comparison to existing similar developments. The findings indicate that:

1. The use of a proposed buffering strategy which enforces class-wise data reordering has improved the individual accuracy of the *eClass* models by 4.05% and by 4.82% on the PID and DERM datasets, respectively. The algorithmic complexity of the buffering method is lesser than that of learning the consequents of the diagnostic rules.
2. Under performance comparisons, the improved *eClass* models outperformed all the other incremental classification methods but ranked after some offline methods, in terms of accuracy. However, most offline methods have exponential complexity and thus they significantly underperform in processing substantial online data. Thus they would fail to process enormously large and growing medical datasets.
3. The improved *eClass* models can evolve more precise diagnostic rules which can be denoted by lingual IF-THEN statements. Therefore, they are human comprehensible and can have positive impact on enhancing the current traditional diagnostic process. For example, an expert can subsequently teach such a model from the results of his own diagnosis. The rules will constantly adapt to any impending expansion of the problem domain and could be used to provide information for next samples.

Future work will focus on reducing the complexity of *eClass*. Its complexity is a square of the number of input dimensions. Therefore the application of feature selection techniques, as recently studied by Ghazavi and Liao [35], seems promising to reduce the complexity but with unknown effects on the model accuracy.

Conflict of interest statement

The authors of this document declare that the related study has been conducted for research purposes. There exists a UK patent application (GB-0621734.3, 1 November 2006) which restricts the use of evolvable fuzzy rule-based classifiers from commercial use.

Acknowledgements

The authors would like to thank Plamen Angelov and Xiaowei Zhou of the University of Lancaster in UK for their contribution of the *eClass* algorithm.

References

[1] Steimann F. On the use and usefulness of fuzzy sets in medical AI. Artificial Intelligence in Medicine 2001;21:131–7.

[2] Zadeh LA. Biological application of the theory of fuzzy sets and systems.. In: Proceedings of international symposium on biocybernetics of the central nervous system. Boston, USA: Little Brown and Co.; 1969. p. 199–212.

[3] Kuncheva LI, Steimann F. Fuzzy diagnosis (editorial). Artificial Intelligence in Medicine 1999;16(2):121–8.

[4] Goncalves LB, Velasco MMBR, Pacheco MAC, De Souza FJ. Inverted hierarchical neuro-fuzzy BSP system: a novel neuro-fuzzy model for pattern classification and rule extraction in databases. IEEE Transactions on Systems Man and Cybernetics (Part C) 2006;36(2):236–48.

[5] Loo CK, Rao MVC. Accurate and reliable diagnosis and classification using probabilistic ensemble simplified fuzzy ARTMAP. IEEE Transactions on Knowledge and Data Engineering 2005;17(11):1589–93.

[6] Watkins A, Timmis J, Boggess L. Artificial immune recognition system (AIRS): an immune inspired supervised machine learning algorithm. Genetic Programming and Evolvable Machines 2004;5:291–317.

[7] Arulampalam G, Bouzerdoum A. Application of shunting inhibitory artificial neural networks to medical diagnosis. In: Proceedings of the 7th Australian and New Zealand intelligent information systems conference; 2001. p. 89–94.

[8] Demiroz G, Govenir HA, Ilter N. Learning differential diagnosis of erythemato-squamous diseases using voting feature intervals. Artificial Intelligence in Medicine 1998;13:147–65.

[9] Zhou X, Angelov P. Autonomous visual self-localization in completely unknown environment using evolving fuzzy rule-based classifier. In: Proceedings of IEEE symposium on computational intelligence in security and defense applications; 2007. p. 131–8.

[10] Zhou X, Angelov P. Real-time joint landmark recognition and classifier generation by an evolving fuzzy system. In: Proceedings of IEEE world congress on computational intelligence; 2006. p. 6314–21.

[11] Xydeas C, Angelov P, Chiao S, Reoullas M. Advances in classification of EEG signals via evolving fuzzy classifiers and dependant multiple HMMs. Computers in Biology and Medicine 2005;36(10):1064–83.

[12] Lekkas S, Mikhailov L. Breast cancer diagnosis based on evolvable fuzzy classifiers and feature selection. In: Allen T, Ellis R, Petridis M, editors. Proceedings of the 28th international conference on innovative techniques and applications of artificial intelligence. Cambridge, UK: Springer; 2008. p. 185–95.

[13] Holt R, Hanley N. Essential endocrinology and diabetes. Malden, USA: Blackwell Publishing; 2006.

[14] Knowler WC, Bennett PH, Bottazzo GF, Doniach D. Islet cell antibodies and Diabetes Mellitus in Pima Indians. Diabetologia 1979;17(3):161–4 [Springer-Verlag, Heidelberg, Berlin].

[15] Angelov P, Filev D. Simpl_eTS: a simplified method for learning evolving Takagi-Sugeno fuzzy models. In: Proceedings of the 11th IEEE international conference on fuzzy systems; 2005. p. 1068–73.

[16] Angelov P, Zhou X, Klawonn F. Evolving fuzzy rule-based classifiers. In: Proceeding of the IEEE symposium on computational intelligence applications in image and signal processing; 2007.p. 220-5.

[17] Angelov P, Zhou X, Filev D, Lughofer E. Architectures for evolving fuzzy rule-based classifiers. In: Proceedings of IEEE international conference on systems, man and cybernetics; 2007. p. 2050–5.

[18] Angelov P, Zhou X. Evolving fuzzy rule-based classifiers from data streams. IEEE Transactions of Fuzzy Systems Special Issue on Evolving Fuzzy Systems 2008;16(6):1462–75.
[19] Chiu SL. Extracting fuzzy rules for pattern classification by cluster estimation. In: Proceedings of the 6th international fuzzy systems association world congress; 1995. p. 1–4.
[20] Pima Indians Diabetes dataset. Available from: http://archive.ics.uci.edu/ml/machine-learning-databases/pima-indians-diabetes/pima-indians-diabetes.data. Accessed: 1st of May, 2008.
[21] Dermatology dataset. Available from: http://archive.ics.uci.edu/ml/machine-learning-databases/dermatology/dermatology.data. Accessed: 1st of May, 2008.
[22] Cornuejols A. Getting order independence in incremental learning.In: Proceedings of European conference on machine learning, Vienna, Austria. Lecture Notes in Artificial Intelligence 1993;667:196–212 [Springer-Verlag, Berlin].
[23] Aggarwal CC, Han J, Wan J, Yu PS. On demand classification of data streams. In: Proceedings of the international conference in knowledge discovery data mining; 2004. p. 503–8.
[24] Luukka P, Leppalampi T. Similarity classifier with generalized mean applied to medical data. Computers in Biology and Medicine 2006;36:1026–40.
[25] Sahan S, Polat K, Kodaz H, Gunes S. The medical applications of attribute weighted artificial immune system (AWAIS): diagnosis of hepatitis and diabetes diseases. Lecture Notes in Computer Science 2005;3627:456–68 [Springer, Berlin].
[26] Kayaer K, Yildirim T. Medical diagnosis on Pima Indian diabetes using general regression neural networks. In: Proceedings of international conference on artificial neural networks neural information processing; 2003. p. 181–4.
[27] Polat K, Gunes S, Arshlan A. A cascade learning system for classification of diabetes disease: generalized discriminant analysis and least square support vector machine. Expert Systems with Applications 2008;34(1):482–7.
[28] Osman HE, Osamu H. Online incremental random forests. In: Proceedings of international conference on machine vision; 2007. p. 102–6.
[29] Domeniconi C, Gunopulos D. Incremental support vector machine construction. In: Cercone N, Lin T, Wu X, editors. Proceedings of the IEEE international conference on data mining. 2001. p. 589–92.
[30] Park MS, Choi JY. Novel incremental principal component analysis with improved performance. Lecture Notes in Computer Science 2008;5342: 592–601 [Springer, Berlin].
[31] Mencar C, Castellano G, Fanelli AM. Mining diagnostic rules using fuzzy clustering. In: De Oliveira JV, Pedrycz W, editors. Advances in Fuzzy Clustering and its Applications. West Sussex, UK: John Wiley & Sons; 2007.
[32] Chang RKY, Loo CK, Rao MVC. A global *k*-means approach for autonomous cluster initialization of probabilistic neural network. Informatica 2008;32: 219–25.
[33] Hayashi H, Zhao Q. Evolving NNTrees more efficiently. In: Proceedings of IEEE congress on evolutionary computation; 2006. p. 623–8.
[34] Ding H, Pei W, He Z. A multiple objective optimization based echo state network tree and application to intrusion detection. In: Proceedings of IEEE international workshop on VLSI design and video technology; 2005. p. 443–6.
[35] Ghazavi SN, Liao TW. Medical data mining by fuzzy modeling with selected features. Artificial Intelligence in Medicine 2008;43(3):195–206.