Neurocomputing 72 (2009) 3580–3589

Contents lists available at ScienceDirect

Neurocomputing

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

Information-theoretic feature selection for functional data classification ☆

Vanessa Gómez-Verdejo [a,*], Michel Verleysen [b], Jérôme Fleury [c]

[a] Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Avda. Universidad 30, 28911 Leganés, Madrid, Spain
[b] DICE - Machine Learning Group, Université Catholique de Louvain, 3 Place du Levant, B-1348 Louvain-la-Neuve, Belgium
[c] Manufacture Française des Pneumatiques Michelin, Bât F32 - Site de Ladoux, 23 Place des Carmes-Deschaux, 63040 Clermont Ferrand, France

ARTICLE INFO

Available online 21 June 2009

Keywords:
Functional data
Classification
Feature selection
Mutual information

ABSTRACT

The classification of functional or high-dimensional data requires to select a reduced subset of features among the initial set, both to help fighting the curse of dimensionality and to help interpreting the problem and the model. The mutual information criterion may be used in that context, but it suffers from the difficulty of its estimation through a finite set of samples. Efficient estimators are not designed specifically to be applied in a classification context, and thus suffer from further drawbacks and difficulties. This paper presents an estimator of mutual information that is specifically designed for classification tasks, including multi-class ones. It is combined to a recently published stopping criterion in a traditional forward feature selection procedure. Experiments on both traditional benchmarks and on an industrial functional classification problem show the added value of this estimator.

1. Introduction

Modeling data having specific structure properties is an important challenge in data analysis. Structures include trees, functions, multi-level data, graphs, and many others. Functional data, i.e., data that are intrinsically curves (despite being usually known through a finite sampling) form an important class of data, as they are found in many industrial application. In particular, functional data can consist in signals, spectra, hysteresis curves, etc.

In practice, working with functional data means to extract a sufficient number of appropriate characteristics (features) from the functions, and then analyzing the features in a quite traditional way. "Sufficient" and "appropriate" are however vague terms, that need to be defined more precisely in the context of a specific problem. In many contexts, it is not obvious to know a priori which features will be useful for the problem. A possible way of working is, firstly, to create a large set of features, and, in a second step, to select the most useful ones for the problem according to a relevance criterion.

Selecting features is often a need for two main reasons. First, it helps fighting the curse of dimensionality; discarding useless features in a regression or classification problem usually improves the learning performances, by reducing the number of (effective) parameters in the model, and thus reducing its variance. Secondly, identifying relevant features helps the application provider understanding how the model behaves, and which physical features are important. Both reasons become essential when the initial number of features extracted from the curves is very large.

Feature selection relies on two main ingredients: a criterion aimed at measuring how a feature, or a subset of features, is relevant for the problem, and a procedure to search the best feature subset, among all possible subsets that could be extracted from the initial one. These two ingredients are essentially unrelated: most criteria can be combined to most search procedures, resulting in a large number of possible feature selection schemes.

About the search procedure, the number of subsets that can be extracted from the initial set is exponential in the number of initial features. In most cases, this results in the impossibility to test all possible subsets, even when the relevance criterion is simple to compute or estimate. Solutions consist in using forward, backward or a combination of both, procedure, genetic algorithms, heuristics, etc. The purpose of this paper is not to cover the wide variety of search procedures; however, some comments will be given about which families of procedures are more adequate to be used in conjunction with the relevance criterion developed in this paper.

About the relevance criterion, the mutual information (MI), a concept borrowed from information theory, is now widely accepted as an appropriate measure of relevance. The MI feature selection criterion [2] has the advantage, over the correlation measure, of being able to measure nonlinear relations between variables and, besides, to evaluate the usefulness of a group of variables instead of a single one. However, its calculation requires to have the data probability distribution, information that is

☆ Expanded version of a communication presented at the IWANN 2007 [12]. This work was supported in part by the MEC Pjt. TEC2008-02473/TEC and the CM Pjt. S-0505/TIC/0223.

* Corresponding author.
E-mail address: vanessa@tsc.uc3m.es (V. Gómez-Verdejo).

0925-2312/$ - see front matter © 2009 Elsevier B.V. All rights reserved.
doi:10.1016/j.neucom.2008.12.035

usually unknown; thus, probability distribution estimators, such as [8,10,22], or more efficient methods that directly estimate the MI must be used. Among the MI estimators that can be found in the literature [2,5,20,24], the estimator proposed by Kraskov et al. in [16], based on K-nearest neighbors distances, has been widely employed for its data efficiency (working with low number of samples) and its high estimation performances.

However, this estimator was originally proposed for regression problems. When it is used for classification tasks, it requires to code the classes with numerical values. This has two limitations: first, the estimator could be made more efficient by directly integrating the fact that outputs are discrete values; secondly, in multi-class problems, it is necessary to design an appropriate numerical coding of classes.

This paper introduces a new MI estimator derived from Kraskov's one, dedicated to classification problems. This dedicated estimator is simpler to use than the general one, and it is adapted to multi-class problems. Using this estimator, a procedure to select features in a functional, multi-class classification application is described; this procedure uses a recently published criterion to stop the search procedure in a sound statistical way.

The following of this paper is organized as follows. Section 2 covers state-of-the-art concepts related to feature selection. It describes the use of the mutual information as relevance criterion for (sets of) features, and traditional ways to estimate it. It also briefly discusses search procedures, including the use of a sound statistical criterion to stop it. It is not the purpose of Section 2 to cover all aspects and recent developments in feature selection; only concepts necessary for the following developments, or other methods that will be used for comparison in the experimental part of the paper, are covered. Section 3 then formulates the proposed MI estimator for classification problems. Section 4 describes a possible procedure to select features from functional data. Section 5 gathers all experimental aspects of the paper. Firstly, it shows the usefulness of the adopted stopping criterion in the search procedure; secondly, it analyzes how the proposed MI estimator improves the classification performances on some traditional, small-size benchmarks; thirdly, it illustrates the procedure on an industrial, functional classification examples. Finally, Section 6 concludes the paper.

2. Feature selection

This section introduces common concepts about feature selection. The mutual information criterion is developed, together with traditional ways to estimate it. Then, standard procedures to search among the possible feature subsets are briefly introduced, and arguments are given to prefer some families of methods, when the MI criterion is used. Finally, a recently published statistical criterion to stop the search procedure is reminded. This section does not include all the state-of-the-art about feature selection; it emphasizes on the concepts necessary to introduce, in Section 3, a new MI estimator for classification problems, and on the methods used in the experimental part of the paper.

2.1. The ingredients of the feature selection process

Feature selection is only possible once an initial set of features is defined. When the original data consist in curves (or functions), the first step is thus to extract features from the curves. The basic principle is that a set of features with maximal content of information must be created. This means, in a generic way, to take basic features such as rough measurements on the data (sampled curves), and to add all features that *could* be relevant for the problem at hand. Traditional possibilities include functional approaches such as the coefficients of splines or other approximators of the curves, similar extraction of the numerical (usually first and second) derivatives of the curves, the location of maxima and minima, the area under curve, etc. The choice of the features is application dependent. However, the idea here is to select a *sufficient* set, even if the price to pay is to increase the number of features. Variables that *could* (but do not necessarily *will*) play a role in the further analysis process should be taken into account. A too high number of features increases the difficulties related to the curse of dimensionality in the design and learning of the classification algorithm. The problem to eliminate useless or redundant features will however be taken into account in a principled way in the next steps of the methodology.

When an initial set of features exists, the problem consists in selecting those that are most relevant for the classification problem at hand. This process has two goals. First, it reduces the data vector dimensionality, making easier the design of the classifier of the next stage. It is indeed known that any classification method (or, more generally, data analysis learning algorithm) suffers from the *curse of dimensionality*, a term that gathers all phenomena related to the difficulties to learn a process, a classification task, etc., in high-dimensional spaces. In practice, the curse of dimensionality concerns problems related to an excessive number of parameters in a model (therefore the variance of the model can be too large if the number of learning data is limited), to the difficulty to design similarity measures in high-dimensional spaces, to the loss of intuitive and geometrical interpretation in high-dimensional spaces, and many other phenomena.

Second, even when no performance improvement is seen when the number of features is reduced, a further advantage comes from the fact that having a low number of features makes the interpretation of the model easier. If an objective measure of the usefulness of features is available (which is the goal of the feature selection process), practitioners may deduce information about the process, and even drive their next measurement campaigns accordingly. Feature selection may therefore be of high interest, both on the performance and interpretation point of views.

Many different feature selection procedures can be designed; however, all of them require the combination of two key elements: a relevance criterion and a subset search procedure.

The relevance criterion scores a feature or a group of features according to its/their capacity for predicting the output (class). Among the different criteria that can be found in the literature, the mutual information measure is widely used for this purpose. MI measures the amount of information contained in a variable X in order to predict the variable Y. Unlike correlation, MI measures *any* relation between X and Y, not only linear ones; furthermore, the MI concept is directly applicable to groups of variable (i.e., X and/or Y can be vectors instead of scalars), as detailed in the next subsection. The mutual information is a well-defined concept, directly applicable as relevance criterion for feature selection. However, its estimation is difficult, as the MI definition relies on the distributions of the X and Y variables, which are unknown in practice. Estimators are thus needed. Section 2.2 will remind the definition of MI and some widely used estimators.

When a relevance criterion is chosen (together with its estimator), a second element is to choose which feature subsets will be evaluated (before choosing the best one). Indeed if variable X is N-dimensional (in other words, if N initial features are available), there are 2^N-1 non-empty possible subsets. Testing all of them, even when the relevance criterion is easy to evaluate, is impossible for large N. There is thus a need for a greedy procedure to search among a reduced number of subsets, while reducing the risk of missing a potentially interesting one. Section 2.3 will briefly summarize possible search procedures, and

it will emphasize on those that are preferred when a MI estimator is used as relevance criterion.

2.2. Mutual information

The mutual information measures how two variables X and Y are related one to each other. The relation is not restricted to be linear. Let us denote the marginal density probability distributions of X and Y as $p_x(x)$ and $p_y(y)$, respectively, and the joint probability distribution of X and Y as $p_{x,y}(x,y)$. The MI between X and Y is given by [4]

$$I(X,Y) = \int\int p_{x,y}(x,y)\log\frac{p_{x,y}(x,y)}{p_x(x)p_y(y)}. \quad (1)$$

Note that in the above definition, both X and Y can be multi-dimensional random vectors; if X gathers a subset of features (from the initial set) and Y is the class information, $I(X,Y)$ can be directly used to measure the relevance of this subset of features.

The problem of this relevance measure is that it relies on the probability density functions; in any practical application, where only a finite sample of data is known, the probability density functions are unknown. Because of this, the MI value has to be estimated. Many estimators can be used, among them traditional histograms [5] and kernel-based estimators [24]. Nevertheless, the search procedure detailed below will use vectors X of increasing dimension, making these estimators inefficient because of the curse of dimensionality. Therefore, a direct estimation of the MI is often preferred; for example, the MI estimator presented in [16] directly estimates the multi-dimensional MI through nearest neighbor counts. An interesting alternative is to avoid computing the multi-dimensional MI, using instead a criterion that simultaneously maximizes the relevance of adding a feature and minimizes its redundance with the already selected ones. In this specific case the relevance criterion is linked to the (forward) search procedure. The minimum redundancy–maximum relevance (MRMR) method [20], and, in particular, the FCD (F-test correlation difference) [6,7] algorithm use this principle.

The following of this section briefly details three estimators that will be used in the following of this paper: an estimator based on Parzen windows, a second one which directly estimated the MI by nearest neighbor counts, and the FCD.

(1) *MI estimation trough distribution estimation with Parzen windows*: A first possibility to estimate the MI is to first estimate the $p_x(x)$, $p_y(y)$and $p_{x,y}(x,y)$ distributions, and then to plug them in (1). Histograms could be used to estimate the distributions, but Parzen windows provide a smoother, more reliable estimate. The following estimator, denoted $\hat{MI}_{Parzen}(X,Y)$, is presented in [17]; it will be used as reference method in the experimental part of this paper. It is designed to be exclusively applied on classification problems.
Let us consider a multi-class classification problem, where a d-dimensional observation $\mathbf{x}_n$ $(1\le n\le N)$ has to be classified among L classes with class labels y_l $(1\le l\le L)$. Then, the MI is estimated as

$$\begin{aligned}\hat{MI}_{Parzen}(X,Y) &= H(Y) - H(Y/X)\\ &= -\sum_{l=1}^{L} p(y=y_l)\log(p(y=y_l))\\ &\quad + \sum_{n=1}^{N}\frac{1}{N}\sum_{l=1}^{L} p(y=y_l|\mathbf{x}_n)\log(p(y=y_l|\mathbf{x}_n)), \end{aligned} \quad (2)$$

where $H(\cdot)$ is the entropy estimator. Due to the fact that Y is a discrete random variable, we have that

$$p(y=y_l) = n_l/N, \quad l=1,\ldots,L, \quad (3)$$

where n_l is the number of observations in class l and L the number of classes. The conditional probability $p(y=y_l|\mathbf{x}_n)$ results from a Parzen estimator with Gaussian window

$$p(y=y_l|\mathbf{x}_n) = \frac{\sum_{i\in y_l}\exp-((\mathbf{x}_n-\mathbf{x}_i)^T\Sigma^{-1}(\mathbf{x}_n-\mathbf{x}_i)/2h^2)}{\sum_{l=1}^{L}\sum_{i\in y_l}\exp-((\mathbf{x}_n-\mathbf{x}_i)^T\Sigma^{-1}(\mathbf{x}_n-\mathbf{x}_i)/2h^2)}. \quad (4)$$

In this equation, h is the window width and Σ the data covariance matrix. In Section 4, and following the recommendations from [17], h has been set to $1/\log(N)$ and Σ has been defined as a diagonal matrix where the $[\Sigma]_{j,j}$ element is given by $2h\sigma_j$, with σ_j the standard deviation of the j-th component of the random variable X.

(2) *Kraskov's estimator of mutual information*: Another way to proceed is to directly estimate the MI, without first estimating the distributions. Intuitively, this can be done with nearest neighbors. Let us consider joint observations (x_n,y_n), $1\le n\le N$, of the random variable (X,Y). When choosing a specific observation, if its neighbors defined in the X space (only) and in the Y space (only) correspond to the same observations, this means that there is a strong relation between X and Y (thus a high MI). According to this intuitive principle, Karskov's estimator [16] works as follows.
It is known that the MI between X and Y can be obtained by means of

$$MI(X,Y) = H(X) + H(Y) - H(X,Y). \quad (5)$$

Therefore, if we had a good entropy estimator, we would only have to replace it in the above expression to obtain an MI estimator. This is the idea followed in [16] to propose two MI estimators; concretely, as starting point, the Kozachenko–-Leonenko estimator for differential Shannon entropy [15] is used. This estimator is defined as

$$\hat{H}(X) = -\psi(K) + \psi(N) + \log c_d + \frac{d}{N}\sum_{n=1}^{N}\log\varepsilon(n,K), \quad (6)$$

where $\psi(\cdot)$ is the digamma function, K is the number of nearest neighbors (a parameter of the algorithm), N is the number of samples in the data set, c_d is the volume of a unitary ball, d is the dimensionality of X and $\varepsilon(n,K)$ is twice the distance from x_n to its Kth neighbor. After some mathematical manipulations, two different MI estimators are derived (see [16] for a detailed explanation)

$$\hat{MI}^{(1)}(X,Y) = \psi(N) + \psi(K) - \frac{1}{N}\sum_{n=1}^{N}[\psi(\tau_x(n)+1) + \psi(\tau_y(n)+1)], \quad (7)$$

$$\hat{MI}^{(2)}(X,Y) = \psi(N) + \psi(K) - \frac{1}{K} - \frac{1}{N}\sum_{n=1}^{N}[\psi(\tau_x(n)) + \psi(\tau_y(n))], \quad (8)$$

where $\tau_x(n)$ and $\tau_y(n)$ are the number of neighbors within distance $\varepsilon(n,K)/2$ from x_n and y_n, respectively. Although both versions show similar results, the implementation of the second one can be found in the mutual information least-dependent component analysis (MILCA) toolbox [1]. In the following of this paper, this implementation will be used and denoted as $\hat{MI}_{MILCA}$.
As it does not require the estimation of the distributions, a difficult task when the dimension (of X) increases, the $\hat{MI}_{MILCA}$ estimator is expected to perform better than the previous one.

Furthermore, it does not make any assumption on Y, so it can be applied to both regression and classification tasks. However, in classification problems, it presents two inconvenients: (1) no use is thus made of the important information that Y is discrete; (2) the global efficiency of the method is reduced because, in the method itself, collisions usually happen during the nearest neighbor search in the Y space. Section 3 will propose a new estimator derived from $\hat{MI}_{MILCA}$, but adapted to classification tasks.

(3) *F-test correlation difference (FCD)*: Another idea is to avoid any estimation (distribution or MI) in a high-dimensional space. This idea naturally brings an advantage over the previous estimators what concerns the curse of dimensionality, the price to pay being that it is not exactly the MI that is estimated, but a related relevance measure. A way of doing this is to combine two relevance criteria, one maximizing the relevance of a feature added to the already existing subset, and another one minimizing the redundancy with the already selected features, such as in the minimum redundancy–maximum relevance approach [20]. When the features are discrete variables, the relevance and redundancy measures are usually the mutual information values; however, when the features are continuous, the F-statistic is employed as maximum relevance score and the Pearson correlation coefficient as minimum redundancy condition.

This last situation corresponds to the FCD algorithm [6,7] that carries out a forward search maximizing the difference between the two criteria; i.e., in tth iteration of the algorithm, when the set of $t-1$ features $S=\{X_1^{sel},\ldots,X_{t-1}^{sel}\}$ have been already selected, the next selected feature, X_t^{sel}, is chosen as

$$X_t^{sel}=\underset{X_j}{\operatorname{argmax}}\left\{F(X_j,Y)-\frac{1}{t-1}\sum_{i\in S}|c(X_j,X_i)|\right\}, \tag{9}$$

where $c(X_j,X_i)$ is the Pearson correlation value and $F(X_j,Y)$ is given by

$$F(X_j,Y)=\frac{\frac{1}{L-1}\sum_{l=1}^{L}n_l(\mu_l-\mu)}{\frac{1}{N-L}\sum_{l=1}^{L}(n_l-1)\sigma_l^2}. \tag{10}$$

In the last equation, μ_l and σ_l^2 are, respectively, the mean and variance of the jth coordinate of the data belonging to the lth class and μ is the overall data mean.

The FCD algorithm, which combines the relevance criterion and a forward search procedure, will be used for comparisons in the experimental part of this paper.

2.3. Search procedure

The second ingredient of the feature selection process is the search procedure that allows us to find the most adequate subset of features (the ones that achieve the maximum MI value) without evaluating all the 2^N-1 possible subsets among N initial features; note that this evaluation would be unfeasible for large N from a computational time point of view.

All standard search procedures can be used, such as the forward search (starting from an empty set and adding features one by one according to the criterion), the backward search (starting from the full set and removing features one by one according to the criterion), or any combination of them (allowing some removals in a forward search or some additions in a backward search). Genetic algorithms can even be used too. The literature contains a high number of references about search procedures, applied in many contexts; see for example [23] in the context of time-series prediction.

An important comment about these options concerns the dimension of the feature subsets at each step of the procedure. In the forward case the dimension of X starts from scratch, and never exceeds the final number of selected features. In the backward case however, its starts from N, the dimension of the initial set of features. If one of the two first estimators ($\hat{MI}_{Parzen}$ or $\hat{MI}_{MILCA}$) or an extension of them is used, the estimator itself is subject to the curse of dimensionality. In these cases, the backward search should definitely be avoided.

In this paper a forward–backward algorithm is used, inspired from [21]. This algorithm works in the following way:

(1) The first selected feature is the one, from the set of all the original features $\{X_1,\ldots,X_N\}$, that maximizes the MI with the output variable Y, i.e.,

$$X_1^{sel}=\underset{X_j}{\operatorname{argmax}}\,\hat{MI}(X_j,Y),\quad 1\le j\le N, \tag{11}$$

where $\hat{MI}(X,Y)$ represents the MI estimator of $MI(X,Y)$ and X_1^{sel} denotes the first selected feature.

(2) The next components must be selected so that the MI between the output Y and the selected set of variables is maximized; in other words, if the algorithm is in t th step and $S=\{X_1^{sel},\ldots,X_{t-1}^{sel}\}$ is the subset of features that have been selected up to step $t-1$, the next selected feature, X_t^{sel}, is chosen as

$$X_t^{sel}=\underset{X_j}{\operatorname{argmax}}\,\hat{MI}(\{S,X_j\},Y),\quad 1\le j\le N,\ \ X_j\notin S. \tag{12}$$

(3) After adding X_t^{sel}, the backward procedure consists in checking if there is an MI increment when removing one by one any of the previous selected components $(X_1^{sel},\ldots,X_{t-1}^{sel})$. If the removal of several variables (one by one) leads to increasing MI, the one (X_t^{rem}) that produces the largest increment is removed, i.e.,

$$X_t^{rem}=\underset{X_j^{sel}}{\operatorname{argmax}}\,\hat{MI}(\{X_1^{sel},\ldots,X_{j-1}^{sel},X_{j+1}^{sel},\ldots,X_{t-1}^{sel}\},Y),\quad 1\le j\le t,$$
$$\text{if}\quad \hat{MI}(\{X_1^{sel},\ldots,X_{j-1}^{sel},X_{j+1}^{sel},\ldots,X_t^{sel}\},Y)>\hat{MI}(\{X_1^{sel},\ldots,X_t^{sel}\},Y). \tag{13}$$

The above algorithm checks at each step if the addition (or removal) of a feature increases the mutual information. Naturally, it is stopped when the MI cannot increase anymore. The problem with this way of working is that it is not theoretically sound, at least what concerns the forward step (adding a feature). Indeed simple developments from the MI definition show that the MI cannot decrease when a feature is added to the existing subset; at worst, the MI value remains identical. The fact that the forward algorithm stops in practice is thus purely related to an artifact of the estimator; for example [9] shows this effect when the $\hat{MI}_{MILCA}$ estimator is used.

To solve this problem and to stop adequately the forward search, [9] suggests to use a permutation test [11] as follows. Let us consider that we are in the tth iteration of the forward search, where $S=\{X_1^{sel},\ldots,X_{t-1}^{sel}\}$ is the subset of already selected features, and X_t^{cand} is the candidate to be added to S, i.e.,

$$X_t^{cand}=\underset{X_j}{\operatorname{argmax}}\,\hat{MI}(\{S,X_j\},Y),\quad 1\le j\le N,\quad X_j\notin S.$$

Then, we can create a random permutation of X_t^{cand} (without permuting the corresponding values of Y), denoted $\tilde{X}_t^{cand}$, and evaluate $\hat{MI}(\{S,\tilde{X}_t^{cand}\},Y)$. If $\hat{I}(\{S,X_t^{cand}\},Y)$ is significantly higher than $\hat{MI}(\{S,\tilde{X}_t^{cand}\},Y)$, we can claim that X_t^{cand} provides new information about Y and has to be added to S; in the opposite case, we can stop the forward search. This procedure has two advantages. First, it relies on comparisons between MI estimations on variables (X and $\tilde{X}$) of the same dimension; the bias of the estimator is thus reduced. Secondly, a standard statistical test may be used to check the significance of the difference between the two estimated MI. Details on the procedure may be found in [9]. An experimental validation of this stopping criterion is included in the experimental section of this paper.

After a subset of features has been selected, the problem becomes a traditional classification task. Therefore, any nonlinear classification tool [13], such as a multi-layer perceptron (MLP), a radial-basis function network (RBFN), a support vector machine (SVM), etc. could be used. The question to know which model has to be preferred exceeds the scope of this work.

An important comment about the feature selection procedure is that knowledge about the variables can (optionally) easily be incorporated between the feature selection and classification stages. In particular, one might be tempted to replace some of the selected variables by other ones having more physical interpretation, if the price to pay in terms of classification performance is low. More specifically, the feature selection results in a single set of variables. Nevertheless, nothing prevents the user to test other sets, i.e., to evaluate their information content by using the same MI estimator as above. Of course, all combinations of variables cannot be tested, otherwise the benefit of the greedy (forward–backward) feature selection would be lost. However, one can, for example, measure the MI between all pairs of single variables, and replace one by one non-interpretable variables by interpretable ones, choosing the latter as to maximize the MI with the deleted variable (they are more or less equivalent) and in the meantime minimize the loss of MI between the new set and the output (the replacement does not change a lot the information content of the set). The choice is intentionally left to the expert user, who is able to judge if the price to pay (in terms of performance decrease) is acceptable to gain interpretability. The use of the MI criterion makes this last procedure possible and measurable in an objective way, if needed by the application, without making it mandatory. It can however be very useful in some industrial context, where the need of interpreting features is at least as important as the need for classification performances.

3. A specific MI estimator for classification problems

The $\hat{MI}_{MILCA}$ estimator does not use the fact that Y is a discrete variable in a classification context. As detailed above, besides being suboptimal, the estimator could be faced to collision problems in the nearest neighbor search when it is used with discrete variables. To improve the efficiency of the MI estimation, in the case of a classification task, the following estimator is proposed. It is valid both for two-classes and multi-classes problems.

Let us consider the multi-class classification problem defined in Section 2.1 and remember that due to Y is a discrete random variable, we can use the information provided by the training data set and estimate its probability distribution as $p(y=y_l)=n_l/N$, $l=1,\ldots,L$. Then, we can define the MI by means of the conditional entropies

$$MI(X,Y)=H(X)-H(X/Y)=H(X)-\sum_{l=1}^{L}p(y=y_l)H(X/Y=y_l). \quad (14)$$

To obtain the desired expression, the entropy estimator (6) must be substituted in (14), i.e.,

$$\hat{MI}(X,Y)=\hat{H}(X)-\sum_{l=1}^{L}p(y=y_l)\hat{H}(X/Y=y_l) \quad (15)$$

$$\begin{aligned}&=-\psi(K)+\psi(N)+\log c_d+\frac{d}{N}\sum_{n=1}^{N}\log\varepsilon(n,K)\\&\quad-\sum_{l=1}^{L}p(y=y_l)\left[-\psi(K)+\psi(n_l)+\log c_d+\frac{d}{n_l}\sum_{n\in y_l}\log\varepsilon_l(n,K)\right] \quad (16)\\&=-\psi(K)+\psi(N)+\log c_d+\frac{d}{N}\sum_{n=1}^{N}\log\varepsilon(n,K)-\psi(K)-\log c_d\\&\quad-\sum_{l=1}^{L}p(y=y_l)\left[\psi(n_l)+\frac{d}{n_l}\sum_{n\in y_l}\log\varepsilon_l(n,K)\right]. \quad (17)\end{aligned}$$

In these equations $\varepsilon(n,K)$ is twice the distance form sample x_n to its Kth neighbor considering all the training data set, whereas $\varepsilon_l(n,K)$ limits the set of possible neighbors to the data from class y_l.

Taking into account that $p(y=y_l)=n_l/N$, a few manipulations lead to

$$\begin{aligned}\hat{MI}(X,Y)&=\psi(N)-\frac{1}{N}\sum_{l=1}^{L}n_l\psi(n_l)\\&\quad+\frac{d}{N}\left[\sum_{n=1}^{N}\log\varepsilon(n,K)-\sum_{l=1}^{L}\sum_{n\in y_l}\log\varepsilon_l(n,K)\right]. \quad (18)\end{aligned}$$

In practice there is no optimal way to select the value of K, except by time-consuming cross-validation. It is therefore suggested to average the estimator over a range of possible values of K, limiting the risk of strong under- or over-fitting. As the proposed MI estimator has been implemented using the approximate near neighbor library [18] (which returns for a fixed value of K the distances of each data point from the first to the Kth neighbor), this can be done without a significant computational cost increase. The final MI estimator, which will be denoted as $\hat{MI}_{Classif}$, is then

$$\begin{aligned}\hat{MI}_{classif}&=\psi(N)-\frac{1}{N}\sum_{l=1}^{L}n_l\psi(n_l)\\&\quad+\frac{d}{N(K_{max}-K_{min}+1)}\left(\sum_{k=K_{min}}^{K_{max}}\left[\sum_{n=1}^{N}\log\varepsilon(n,k)-\sum_{l=1}^{L}\sum_{n\in y_l}\log\varepsilon_l(n,k)\right]\right), \quad (19)\end{aligned}$$

where K_{min} and K_{max} determine the range of K values. Suggested values for K_{min} and K_{max} are 4 and 12, respectively. Note that if the $\hat{MI}_{MILCA}$ estimator was used instead of the $\hat{MI}_{Classif}$ one, $\tau_x(n)$ and $\tau_y(n)$ would have to be calculated for each value of K what would significantly increase the computational cost.

Finally, it is important to remark that, unlike $\hat{MI}_{MILCA}$, this estimator does not need to codify the output variable, since it takes advantage of the fact that outputs are discrete and, therefore, it avoids to employ an inappropriate numerical class coding.

Table 1
Evaluation of the $\hat{MI}_{MILCA}$ and the proposed $\hat{MI}_{Classif}$ estimators embedded in a forward search with two possible stopping criteria: (1) MI maximum value and (2) permutation test.

Problem	$\hat{MI}_{MILCA}$		$\hat{MI}_{Classif}$	
	MI maximum	Permutation test	MI maximum	Permutation test
Breast	97.07% {3, 7, 10}	96.59% {3, 7}	96.59% {3, 7, 6}	96.59% {3, 7}
Glass	47.69% {4, 7, 8}	52.31% {4, 7, 8, 5, 3, 1, 6, 9, 2}	53.83% {2, 4, 7}	53.83% {2, 4, 7, 3}
Letter	79.02% {13, 15, 8, 9, 11, 10, 12, 16, 7}	85.08% {13, 15, 8, 9, 11, 10, 12, 16, 7, 6, 14, 2, 5, 4, 1, 3}	13.23% {11}	85.01% {11, 12, 10, 13, 15, 8, 9, 16, 6, 14, 7, 2, 5, 1, 3, 4}
Pima	74.78% {2, 8, 3}	76.09% {2, 8, 3, 6}	74.78% {7, 2}	74.78% {7, 2, 8}
Wine	92.45% {7, 10, 13}	98.11% {7, 10, 13, 1, 6, 12, 5, 11, 9, 3, 4}	92.45% {7, 10, 13}	100% {7, 10, 13, 4, 1, 3, 9, 11}
Wave	82.80% {7, 11, 15, 12, 17, 10, 13, 16}	83.60% {7, 11, 15, 12, 17, 10, 13, 16, 6, 9}	79.80% {7, 11, 17, 16, 13}	84.67% {7, 11, 17, 16, 13, 10, 12, 6, 8, 15, 9}

The classification accuracies obtained with a linear SVM and final subsets of selected features are presented.

4. Experiments

This section gathers the experimental results obtained with the criterion and procedures described in this paper. It is structured in three parts. The first, preliminary one, confirms by simple experiments the interest for the forward–backward search stopping criterion described in Section 2.3; these results corroborate other experimental results that can be found in [9]. The second part shows classification results obtained after feature selection, both with the classification-specific MI criterion proposed in this paper and with other existing criteria. This comparative study uses standard (non-functional) classification benchmarks. The third part addresses a real-world industrial problem of hysteresis curve classification. Through this last example it is shown how to design a features selection strategy in several stages, when application-specific information about the features is available.

4.1. Forward search stopping criterion

As detailed in Section 2.3, a forward or forward–backward search using mutual information as criterion is usually stopped when the MI estimation does not increase anymore. However the decrease of the MI estimation is purely due to the bias and variance of the estimator, making this strategy neither sound nor optimal. A stopping criterion based on the permutation test was summarized in Section 2.3, based on a procedure described in [9].

To show the necessity of this stopping criterion when the proposed $\hat{MI}_{Classif}$ is employed, the classification performances obtained on standard benchmarks from the UCI machine learning repository [19] are reported here. A forward search with both the $\hat{MI}_{MILCA}$ and $\hat{MI}_{Classif}$ estimators was used. The classification method is a linear SVM. Although not optimal for all considered problems, this algorithm has been chosen for its simplicity and lack of tuning parameters (such as kernel choice in nonlinear SVM); best classification performances are not searched for, but only comparisons at the level of MI estimators and stopping criteria.

Concretely, six benchmark problems from the UCI repository have been employed: *Breast*, *Glass*, *Letter*, *Pima*, *Wine*, and *Wave*. Because these problems do not have a predefined test partition, to test the performance of the classifiers, we have splitted the original training set in a 70%/30% partitions to train and test our algorithms, respectively.

Table 1 shows the classification results obtained when the forward search is stopped at the maximum of MI, and with the permutation test. The table shows the classification accuracy and the final subsets of selected features.

At the light of the results from Table 1, we can observe in the case of the $\hat{MI}_{MILCA}$ estimator, that the permutation test provides a systematic accuracy improvement over the maximum MI value criterion, except for *Breast* where a slight performance deterioration is observed. In similar way, we can observe this improvement when the permutation test is employed with the proposed $\hat{MI}_{Classif}$ estimator; however, in this case, two different effects are presented:

- First, in *Wine*, *Wave*, and *Letter*, the permutation test significantly improves the classification accuracy compared to the MI maximum value criterion; the performance improvements are around 5% in *Wave*, 8% in *Wine* and even 70% in *Letter*.
- Secondly, in *Breast*, *Glass*, and *Pima*, the permutation test results in the same accuracy than the MI maximum value criterion, although the selected features subsets are slightly different. It should be noted that the number of features that differ in both cases is small (1 in each of the three cases), and this is due to the fact that classification results obtained by any filter method can never be optimal (the criterion, MI in this case, never optimizes the true classification performances).

According to these results confirming the superiority of the permutation test criterion, this way of stopping the forward–backward search will be used in the next experiments, when the $\hat{MI}_{MILCA}$ or $\hat{MI}_{Classif}$ estimator is used.

4.2. Classification performances on standard classification benchmarks

This section evaluates the classification accuracies obtained after feature selection with the proposed $\hat{MI}_{Classif}$ estimator, and with several other estimator described in Section 2.2: $\hat{MI}_{Parzen}$ and $\hat{MI}_{MILCA}$ with forward search and the FCD algorithm; both $\hat{MI}_{Classif}$

Table 2
Evaluation of the performance achieved by different feature selection methods: with all the original features, $\hat{MI}_{Parzen}$, FCD, $\hat{MI}_{MILCA}$, and $\hat{MI}_{Classif}$.

Problem	All features	$\hat{MI}_{Parzen}$	FCD	$\hat{MI}_{MILCA}$	$\hat{MI}_{Classif}$
Breast					
Accuracy (%)	98.05	97.07	85.85	96.59	96.59
# Features	10	3	2	2	2
Glass					
Accuracy (%)	52.31	53.83	53.85	52.31	53.83
# Features	9	6	4	9	4
Letter					
Accuracy (%)	85.08	80.15	85.08	85.08	85.08
# Features	16	12	16	16	16
Pima					
Accuracy (%)	77.39	77.83	65.22	76.09	74.78
# Features	8	5	3	4	3
Wine					
Accuracy (%)	96.23	92.45	90.57	98.11	100
# Features	13	4	8	11	8
Wave					
Accuracy (%)	85.60	72.47	84.80	83.60	84.67
# Features	40	4	11	10	11

The number of selected features (# features) is shown together with the accuracy achieved by a linear SVM over each problem.

and $\hat{MI}_{MILCA}$ have been used with a forward search and the permutation test as stopping criterion. Besides, to corroborate the advantages of the feature selection strategies, the performance obtained when all original features are used are also illustrated. A linear SVM is used for the same reasons as detailed in the previous section.

Table 2 shows the number of selected features by each method (all the original features, $\hat{MI}_{Parzen}$, FCD, $\hat{MI}_{MILCA}$, and $\hat{MI}_{Classif}$), together with the accuracy obtained with a linear SVM. From these results, the following comments may be done:

- The *Glass*, *Pima* or *Wine* problems show the advantages of a feature selection strategy: the feature selection methods provide a significant classification accuracy improvement with regard to employing all original features. Besides, in the remaining problems, we can observe that some of the feature selection methods present similar performance to the all features approach, but using a reduced subset of features; this is the case of $\hat{MI}_{Parzen}$, $\hat{MI}_{MILCA}$, and $\hat{MI}_{Classif}$ in *Breast* or FCD, $\hat{MI}_{MILCA}$, and $\hat{MI}_{Classif}$ in *Wave*. Furthermore, these methods are able to select the complete feature set when all features are necessary to achieve good performances (see the *Letter* problem).
- If we analyze the performance of the proposed $\hat{MI}_{Classif}$ estimator, we can observe that, in five of the six benchmark problems, it presents the best classification accuracy or a result very close to the best. For instance, in *Letter* and *Wine*, the $\hat{MI}_{Classif}$ classification performance is the best one, and in *Breast*, *Glass* and *Wave* it is close to the best; note the 100% classification accuracy obtained in *Wine* with only eight features. On all databases, the selection based on $\hat{MI}_{Classif}$ results in a good compromise between the classification performances and the number of selected features.
- Finally, if we directly compare the use of $\hat{MI}_{Classif}$ with $\hat{MI}_{MILCA}$, we observe that $\hat{MI}_{Classif}$ outperforms $\hat{MI}_{MILCA}$ in three cases and ties in two other ones. This fact is not surprising and points out the advantage obtained by $\hat{MI}_{Classif}$ that makes use of the discrete nature of the dependent variable in a classification problem. In fact, the only problem where $\hat{MI}_{MILCA}$ presents a higher classification accuracy than $\hat{MI}_{Classif}$ is a binary problem (*Pima*), where the superiority of $\hat{MI}_{Classif}$ is not (or less) expected.

4.3. Rigidity and hysteresis curves classification

In this section, the methodology described in this paper is used to solve a specific, industrial problem of hysteresis curve classification.

4.3.1. Original data

In order to know the validity or conformity of an elastomeric material, each sample of material undergoes a deformation process. First an external force is applied over the material; secondly, this force is removed which makes the material come back to its original state. During this process, both the rigidity and the hysteresis of the material are measured for a number (here 23) of deformation values, resulting in two curves, called R and H curves, respectively, in the following. These two curves are used to evaluate the validity of the material.

Fig. 1 shows the typical shapes of these curves. Both the *x*-(deformation) and *y*-axes (rigidity/hysteresis) have been normalized, both for confidentiality reasons, and because this normalization does not influence the further processing. Each graph can be divided into two curves: the forward curve, related to the material deformation when the force is applied and the return curve, linked to the force elimination process. For instance, for the R curve, the forward deformation corresponds to the upper part of the curve and the return deformation corresponds to the lower one; for the H curve, the opposite is observed.

For each material, a data vector with 47 components is measured:

- Component 1: temperature of the experiment.
- Components 2–24: values of the R curve in the material deformation positions (12 in the forward curve and 12 in the backward one, with the point corresponding to the largest deformation in common).
- Components 25–47: values of the H curve in the same deformation positions.

Learning samples belong to three classes and are labeled by experts. The three classes are "conform", "non-conform" and "unknown"; the last class is used when the experts do not know or disagree on the validity of the material and its measure. The training set provided by the industrial experts consists of 633 data (where each data is formed by the above 47 components), with 483 data of the "conform" class, 112 of the "unknown" one and the remaining 83 data of the "non-conform" class. In addition, a test set with 168 data where 119 data belong to the "conform" class, 22 to the "unknown" class and 27 to the "non-conform" class is also provided.

4.3.2. Creation of new features

Since the R and H curves allow experts to know the validity or not of each material, this step consists in creating a new set of features, trying to extract the maximum information from the curves. Previous knowledge on what exact information to extract from the curves is not available. Therefore, according to the guidelines given in Section 2.1, a set of 191 potentially interesting features, which are described in the Table 3, is created.

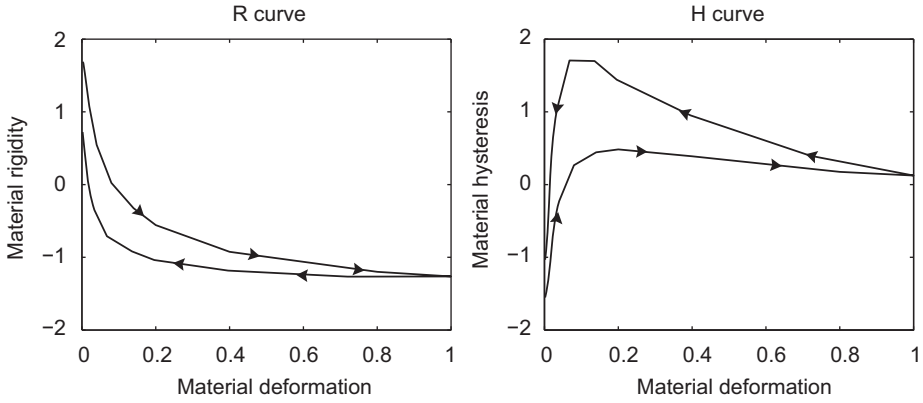

Fig. 1. Example of the normalized rigidity (R) and hysteresis (H) curves.

Table 3
Description of the new set of created features.

Feature description	Feature number	
	R curve	H curve
Temperature of the experiment	1	
Original values	2–24	97–119
Area under the curve	25	120
Numerical first derivatives	26–47	121–142
Widths of the curve	48–58	143–153
Coefficients of fifth degree polynomial	59–70	154–165
Coefficients of linear approximation	71–74	166–169
Coefficients of quadratic approximation	75–80	170–175
Maximum and minimum points	81–88	176–183
Statistical information (moments)	89–96	184–191

Uninteresting ones will be discarded in the next step of the procedure. Basically, these features correspond to derivatives, slopes, widths of the curves at specific deformation positions, area between curves, mimina and maxima, coefficients of the approximation of the curves by polynomials and basic statistical information (mean, standard deviation, skewness and kurtosis). The set of 191 resulting features is summarized in Table 3.

4.3.3. Feature selection

The 191 features are entered into the selection procedure described in Section 2.3. However, because of the high number of features in this application, a hierarchical approach is used to reduce the computation time. The hierarchical approach consists in applying the selection procedure independently over each curve and, besides, over each group of features; the search algorithm is thus divided in three substages, working over a moderate number of features and accelerating the feature selection process. This process has been carried with the following four feature selection methods: the FCD method and the $\hat{M}I_{Parzen}$, $\hat{M}I_{MILCA}$ and $\hat{M}I_{Classif}$ estimators combined with the forward–backward algorithm. Note that contrarily to the preliminary experiments carried out in the previous sections, the goal is here to obtain the best classification accuracy; therefore, the forward–backward search is preferred, because it usually achieves better performances than the classical forward search. To stop the forward–backward algorithm we have used the permutation test with the $\hat{M}I_{MILCA}$ and $\hat{M}I_{Classif}$ estimators and the maximum MI value criterion for $\hat{M}I_{Parzen}$.

To use $\hat{M}I_{MILCA}$ estimator, classes have to be labeled numerically; the "conform", "non-conform" and "unknown" classes have thus been labeled 1, −1 and 0, respectively. Note that despite this problem is not a binary classification one (there are three classes), this is one of the rare examples where ordering class labels is legitimate, as the "unknown" class is clearly between the two other ones for which the true class is known. In all other circumstances (non-ordered classes), the $\hat{M}I_{Parzen}$, $\hat{M}I_{Classif}$ estimators and the FDC method would add a supplementary advantage, as no class ordering or binary coding is necessary.

Table 4
Subset of the selected features with the four feature selection method: $\hat{M}I_{Parzen}$, FCD, $\hat{M}I_{MILCA}$, $\hat{M}I_{Classif}$.

Method	Subset of selected features
$\hat{M}I_{Parzen}$	{121, 142, 144, 186, 190}
FCD	{131, 137, 145, 153, 160, 170, 173, 179, 182}
$\hat{M}I_{MILCA}$	{97, 99, 119, 126, 127, 134, 136, 144, 145, 147, 155, 158, 176, 185}
$\hat{M}I_{Classif}$	{124, 129, 137, 141, 143, 144, 150, 153, 156}

Table 4 shows the features selected by each method after the search procedure. The following can be observed from these results.

- As a result of this process, each method has chosen a different features subset; however in all the cases, the selected features are from the H curve.
- Analyzing the kind of selected features, we can observe that all methods agree to select the features that are related to the H curve derivatives (variables from 121 to 142) and the H curve width (from 143 to 153).
- Furthermore, FCD, $\hat{M}I_{MILCA}$ and $\hat{M}I_{Classif}$ also coincide in choosing some of the fifth degree polynomial coefficients.
- The remaining selected features correspond to statistical information (chosen by $\hat{M}I_{Parzen}$ and $\hat{M}I_{MILCA}$), information about the maximum and minimum points (features 186 and 190 in $\hat{M}I_{Parzen}$ or 185 in $\hat{M}I_{MILCA}$), coefficients resulting from the approximation of the H curve with a parabola (FCD method) or some of the original H curve points (variables 97, 99 and 119 of $\hat{M}I_{MILCA}$).

The final classification results provided by each feature subset will allow to know which kind of information is more useful to carry out the classification task.

4.3.4. Classification

After feature selection, the next step consists in building a classifier; for that purpose, a support vector machine with Gaussian kernels has been employed. The LIB-SVM toolbox [3]

Table 5
Results achieved by the experts' system and the four methods covered in this paper: $\hat{MI}_{Parzen}$, FCD, $\hat{MI}_{MILCA}$, and $\hat{MI}_{Classif}$.

	Proposed system with				Expert's system (%)
	$\hat{MI}_{Parzen}$ (%)	FCD (%)	$\hat{MI}_{MILCA}$ (%)	$\hat{MI}_{Classif}$ (%)	
Global accuracy	79.76	85.12	83.93	87.5	76.13
C_1					
TPR	94.96	97.48	95.8	96.64	81.51
FPR	42.86	18.37	32.65	14.29	8
C_0					
TPR	40.91	45.45	45.45	54.55	65.22
FPR	6.85	6.85	6.16	5.48	21.92
C_{-1}					
TPR	44.44	62.96	62.96	74.07	62.96
FPR	2.13	4.26	1.42	4.26	2.82

In the first row, the global classification accuracy for each system is given; in the following rows, the TPR (True Positive Rate) and FPR (False Positive Rate) linked to each class (C_1: "conformity", C_0: "unknown" and C_{-1}: "no conformity") are shown.

has been used, for its powerful implementation of multi-class classifiers based on error correcting output codes (see [14] for a detailed explanation of its implementation). The kernel dispersion γ (defined as the γ parameter in $K(\mathbf{x}_i, \mathbf{x}_j) = \exp(\gamma\|\mathbf{x}_i - \mathbf{x}_j\|)$) and penalty factor C have been optimized by 10-fold cross-validation on the training set.

4.3.5. Experimental results

The effectiveness of the proposed feature selection methodology and criterion is analyzed in this section. The four solutions using the FDC method and the $\hat{MI}_{Parzen}$ $\hat{MI}_{MILCA}$ and $\hat{MI}_{Classif}$ estimators are compared to a system designed by the experts. Concretely, the experts have selected the set of variables that they consider as most relevant for the problem; next, they have trained and tested a classifier with the same data sets as those used above. Table 5 presents the classification accuracy achieved by the proposed systems and the experts' system.

The application of the proposed general methodology to the classification of the dynamic properties of elastomeric material has provided reduced sets of features that were extracted automatically, without expert knowledge. Table 5 clearly shows that the extracted features are sufficient, and even much better than the ones selected by the experts, to get an acceptable classification accuracy; the proposed system with either $\hat{MI}_{Parzen}$, FCD, $\hat{MI}_{MILCA}$ or $\hat{MI}_{Classif}$ presents a classification accuracy of 79.76%, 85.12%, 83.93% or 87.5% (respectively), whereas the expert's system only achieves an accuracy of 76.13%.

Furthermore, Table 5 shows the improvement obtained by using the MI estimator proposed in this paper; it presents the best classification accuracy with only nine features (less than $\hat{MI}_{MILCA}$ or FCD). Besides the good classification results, the reduced number of selected features also gives an added value to the results in terms of interpretability.

Finally, it is worthy pointing out that the set of features selected with $\hat{MI}_{Classif}$ reveals that aspects related to the derivatives, width of the H curve and some coefficients of the fifth polynomial approximation are sufficient to solve satisfactorily the classification task; conversely, the statistical information chosen by the $\hat{MI}_{Parzen}$ estimator does not seem to be adequate to solve adequately the classification task since the resulting global performance is only 79.76%. Further measurement campaigns could benefit from this result by avoiding the costly acquisition of features that reveal useless at the end.

5. Conclusions

Classifying high-dimensional data often necessitates a feature extraction preprocessing, both to decrease the number of features therefore limiting the effects of the curse of dimensionality, and to help interpreting the model. In the case of curve (function) classification, feature selection is even more important, as there is often no a priori optimal way to extract features from the curves.

In this paper, the feature selection criterion is specifically discussed. It is shown that the mutual information is an adequate criterion to perform feature selection, but that it suffers from the difficulty of obtaining accurate estimations from a finite number of data. Efficient estimators exist, but are designed for regression problems; this paper introduces a new estimator specific for classification tasks, including multi-class ones.

Experiments show that this criterion, embedded in a forward–backward search with a sound stopping criterion, leads to improved performances. The classification accuracy is illustrated with standard classifiers, both on traditional benchmarks and on an industrial curve classification problem. The latter consists in a classification problem of the dynamic properties of elastomeric material; the method proposed in this paper proves to improve both the quality and the interpretation of the results. The proposed estimator of mutual information also reveals to improve the classification performances compared to the same experiments using an estimator not specifically designed for classification tasks.

Although the hysteresis curves classification problem shows the advantages that the proposed estimator can provide, we intend to continue this work applying this new MI estimator to other real applications, such as optical character recognition, speech recognition, image classification, ..., where the high number of input features and the intrinsic multi-class nature of these problems can take the most of the proposed estimator and show its real advantages.

Furthermore, it would be also very useful to analyze the analytical properties of the proposed estimator, for instance, trying to find estimation error bounds; this could help us to understand its behavior and its sensibility with regard to parameter K (number of neighbors) or the number of data.

References

[1] S. Astakhov, P. Grassberger, A. Kraskov, H. Stögbauer, MILCA software: mutual information least-dependant component analysis. Software available at ⟨http://www.klab.caltech.edu/~kraskov/MILCA/⟩.
[2] R. Battiti, Using the mutual information for selecting features in supervised neural net learning, IEEE Transactions on Neural Networks 5 (1994) 537–550.
[3] C. Chang, C. Lin, LIBSVM: a library for support vector machines, 2001. Software available at ⟨http://www.csie.ntu.edu.tw/~cjlin/libsvm⟩.
[4] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley, New York, 1991.
[5] G.A. Darbellay, I. Vajda, Estimation of the information by an adaptive partitioning of the observation space, IEEE Transactions on Information Theory 45 (4) (1999) 1315–1321.
[6] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, in: Proceedings of the Second IEEE Computational Systems Bioinformatics Conference, Stanford, CA, 2003, pp. 523–528.
[7] C. Ding, H. Peng, Minimum redundancy feature selection from microarray gene expression data, Journal of Bioinformatics and Computational Biology 3 (2) (2005) 185–205.
[8] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., Wiley, New York, 2001.
[9] D. François, F. Rossi, V. Wertz, M. Verleysen, Resampling methods for parameter-free and robust feature selection with mutual information, Neurocomputing 70 (2007) 1276–1288.

[10] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, New York, 1990.
[11] P. Good, Permutation Tests, Springer, New York, 1994.
[12] V. Gómez-Verdejo, M. Verleysen, J. Fleury, Information-theoretic feature selection for the classification of hysteresis curves, in: Proceedings of the IWANN 2007, Lecture Notes in Computer Science, Springer, Berlin, vol. 4507, 2007, pp. 522–529.
[13] S. Haykin, Neural Networks: A Comprehensive Foundation, second ed., Prentice-Hall, Englewood Cliffs, NJ, 1999.
[14] T.K. Huang, R.C. Weng, C.J. Lin, Generalized Bradley–Terry models and multi-class probability estimates, Journal of Machine Learning Research 7 (2006) 85–115.
[15] L.F. Kozachenko, N.N. Leonenko, A statistical estimate for the entropy of a random vector, Problems Information Transmission 23 (2) (1987) 9–16.
[16] A. Kraskov, H. Stögbauer, P. Grassberger, Estimating mutual information, Physical Review E 69 (2004).
[17] N. Kwak, C.H. Choi, Input feature selection by mutual information based on Parzen window, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (12) (2002) 1667–1671.
[18] D.M. Mount, S. Arya, Approximate near neighbour (ANN) C ++ library. Software available at 〈http://www.cs.umd.edu/~mount/ANN/〉, 2009.
[19] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI repository of machine learning databases 〈http://www.ics.uci.edu/~mlearn/MLRepository.html〉, University of California, Department of Information and Computer Sciences, Irvine, 1998.
[20] H. Peng, F. Long, C. Ding, Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (8) (2005) 1226–1238.
[21] F. Rossi, A. Lendasse, D. François, V. Wertz, M. Verleysen, Mutual information for the selection of relevant variables spectrometric nonlinear modelling, Chemometrics and Intelligent Laboratory Systems 80 (2006) 215–226.
[22] D.W. Scott, Multivariable Density Estimation: Theory, Practice and Visualization, Wiley, New York, 1992.
[23] A. Sorjamaa, J. Hao, N. Reyhani, Y. Ji, A. Lendasse, Methodology for long-term prediction of time series, Neurocomputing 70 (16–18) (2007) 2861–2869.
[24] R. Steuer, J. Kurths, C.O. Daub, J. Weise, J. Selbig, The mutual information: detecting and evaluating dependencies between variables, Bioinformatics 18 (2) (2002) 231–240.

Vanessa Gómez-Verdejo was born in Madrid, Spain, in 1979. She received the Telecommunication Engineer degree in 2002 from Universidad Politécnica de Madrid; in 2007 she has received the Ph.D. degree from Universidad Carlos III de Madrid, where she is now an assistant professor. Her present research interests are centered in the fields of adaptive signal processing and machine learning, mainly neural networks ensembles and boosting methods.

Michel Verleysen was born in 1965 in Belgium. He received the M.S. and Ph.D. degrees in electrical engineering from the Université catholique de Louvain (Belgium) in 1987 and 1992, respectively. He was an invited professor at the Swiss E.P.F.L. (Ecole Polytechnique Fédérale de Lausanne, Switzerland) in 1992, at the Université d'Evry Val d'Essonne (France) in 2001, and at the Université ParisI-Panthéon- Sorbonne from 2002 to 2007, respectively. He is now a professor at the Université catholique de Louvain, and honorary research director of the Belgian F.N.R.S. (National Fund for Scientific Research). He is an editor-in-chief of the Neural Processing Letters journal, chairman of the annual ESANN Conference (European Symposium on Artificial Neural Networks), associate editor of the IEEE Transactions on Neural Networks journal, and member of the editorial board and program committee of several journals and conferences on neural networks and learning. He is author or co-author of more than 200 scientific papers in international journals and books or communications to conferences with reviewing committee. He is the co-author of the scientific popularization book on artificial neural networks in the series "Que Sais-Je?", in French, and of the "Nonlinear Dimensionality Reduction" book published by Springer in 2007. His research interests include machine learning, artificial neural networks, self-organization, time-series forecasting, nonlinear statistics, adaptive signal processing, and high-dimensional data analysis.

Jérôme Fleury was born in Roanne in 1976. After following statistical formation, he is now working at Michelin Research Complex and in charge of statistical team.