Chemometrics and Intelligent Laboratory Systems 217 (2021) 104423

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

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

A modified entropy-based performance criterion for class-modelling with multiple classes

O. Valencia [a], M.C. Ortiz [b,*], M.S. Sánchez [a], L.A. Sarabia [a]

[a] Dpt. Matemáticas y Computación, Universidad de Burgos, Facultad de Ciencias, Plaza Misael Bañuelos s/n, 09001, Burgos, Spain
[b] Dpt. Química Universidad de Burgos, Facultad de Ciencias, Plaza Misael Bañuelos s/n, 09001, Burgos, Spain

ARTICLE INFO

Keywords:
Sensitivity
Specificity
Class-model
Type I and Type II errors
Entropy
Benchmark

ABSTRACT

The paper presents a new proposal for a single overall measure, the diagonal modified confusion entropy (DMCEN), to assess the performance of class-models jointly computed for several classes, a versatile index regarding sensitivity and specificity, and that supports class weighting.

The characteristics of the proposed figure of merit are illustrated as against other usual performance measures and show how the index is more sensitive to the variations in the class-models than similar published indexes.

Besides, a benchmark value representing a random modelling is also defined for DMCEN to be used as initial level to assess the quality of the built class-models.

Furthermore, systematic comparisons have been conducted by using the degree of consistency *C* and the degree of discriminancy *D* when comparing the proposed DMCEN to the usual total efficiency (a geometric mean between sensitivity and specificity).

Simulations show that, for a hundred thousand sensitivity/specificity matrices with four categories, *C* is almost 0.7 on average, well above the needed 0.5, and there is more than 62% probability that DMCEN detects differences when the total efficiency does not.

Illustration of the application of the index is shown with an experimental data set with four categories.

1. Introduction

Supervised classification problems are at the core of research in different fields including statistics, machine learning, pattern recognition or data mining and application domains as diverse as medicine, finance, quality control and chemistry. In this work we focus on supervised classification based on qualitative patterns, as the objects to be classified belong to a known category or class. Precisely, *K* classes or categories are assumed, conceptually well-defined and intrinsically disjoint. In practice, however, this might not be the case (e.g. areas such as metabolomics or heritage science) because some objects cannot always be assigned with certainty to a known category, or there can be outliers or misclassifications.

In any case, with objects belonging to the different categories (training set), a decision rule is needed, ultimately to assign a new object to one of the categories. These objects are described by several properties, which constitute the input predictor variables used to construct the mathematical or statistical model for the decision.

Basically, there are two different approaches, that traditionally are described in geometrical terms [1]. The first one, the purely discriminant approach, consists of constructing a partition of the space of the input variables, that is, a family of *K* disjoint subsets (with no common elements between one another) whose union is the whole input space. In that case, each object is always unambiguously assigned to one, and only one, of the *K* categories. As such, the performance criteria used for the validation of the computed discriminant models (also called decision rules) are related to their expected classification accuracy, namely, the percentage of correct decisions in prediction. Examples of common purely discriminant methods include linear or quadratic discriminant analysis (LDA, QDA [2]), RDA (regularized discriminant analysis [3]), PLS-DA (partial least squares discriminant analysis [4]), CART (classification and regression trees [5]), or SVM (support vector machines [6]), originally developed for two-class classification [7], generalized to multiclass situation [8] and to the case of unlabeled data [9].

The second approach, the class-modelling approach, also aims at constructing *K* subsets within the input space, one per class (the so-called class-models) but they are not necessarily disjoint sets neither their union is the whole space. Therefore, a new object can be inside one or several

* Corresponding author.
E-mail address: mcortiz@ubu.es (M.C. Ortiz).

https://doi.org/10.1016/j.chemolab.2021.104423
Received 9 July 2021; Received in revised form 6 September 2021; Accepted 11 September 2021
Available online 16 September 2021

class-models or even outside all of them. Performance criteria, also called figures of merit, relative to class-modeling techniques are related to sensitivity and specificity of each class-model. The sensitivity of a class-model refers to its ability for recognizing its own objects (usually estimated as the rate of category objects that are correctly inside the corresponding class-model), whereas the specificity refers to the ability of rejecting foreign objects (estimated as the rate of foreign objects that are correctly outside the class-model). Examples of class-modelling techniques in chemometrics include SIMCA (soft independent models of class analogy [10]), UNEQ (unequal class models or unequal dispersed classes [11]), or an adaptation of SVM known as Support Vector Data Description (SVDD) [12]. Further developments on class-modelling via PLS regression are also conducted in Refs. [13,14]. A review of class-modelling can be found in Refs. [15,16].

There are also discriminant and class-modelling methods that are described in terms of probability distribution functions. In that case, the decision about an object **x** is made based on the probability that **x** belongs to a given class-model. Let p_j denote the probability that **x** belongs to the *j-th* class-model ($j = 1, \ldots, K$). In the discriminant case, the sum of p_j is always one, the probability of the intersection of classes is null, and **x** is assigned to the class with largest p_j. On the contrary, in the class-modelling situation, the probability of the intersection of two or more class-models can be non-null and the sum of all probabilities p_j is not necessarily one.

Another relevant difference is that for discriminant purposes, the training set must contain objects from at least two different categories, whereas in the class-modelling case the focus is on individual classes so that each class is independently modelled, and the methods can be applied when the training set contains objects of a single class, the so-called one-class classifiers [17,18] or compliant class-modelling methods [19].

Whether discriminant or class-modelling methods, the assessment of performance of classifiers has been intensively studied although conclusions are often drawn from empirical research and thus conditioned on the selections made in terms of datasets, experimental procedures and performance metrics. As pointed out in Refs. [20,21], there is no best classifier as such ('no free lunch' theorem) but some classifiers might outperform others in particular domains, for particular tasks and requirements.

Therefore, extensive research has been conducted on performance, mostly on performance metrics, which has resulted in comprehensive lists of measures (continually updating). An experimental study of the behavior of eighteen different performance metrics in several scenarios is conducted in Ref. [22] while in Ref. [23] the invariance properties of several measures are analyzed. Up to nineteen figures of merit are listed in Ref. [24], most of them derived from the usual sensitivity of every individual class-model, pair-wise specificities, efficiency, or total sensitivity, total specificity, and total efficiency of all class-models, including convex combination of individual sensitivities and specificities [25]. More recently, ref. [26] provides a systematic comparison of several global measures of classification together with a proposal of a set of benchmark values based on different random classification scenarios.

Like all values obtained from experimental data, the performance criteria of a classifier are affected by uncertainty caused by both the objects in the training set and the values of the variables. This should be taken into account when evaluating, in practice, the figures of merit of a classifier [20].

Hand [20] also suggests a taxonomy of performance criteria for the binary classification problem which differentiates between problem-based metrics and accuracy-based metrics. The former are designed to meet important requirements of specific domains and applications, such as the speed of the classifier, the time it takes to update, the ability to identify relevant predictor variables or handle particular datasets (large size, incomplete, unbalanced, small-*n*-large-*p*, where usually *n* refers to the number of objects and *p* to the number of variables). Accordingly, metrics are strongly parametrized to be able to include the problem knowledge. On the contrary, the latter type, widely used and almost automatic, is focused on how well the classifier assigns objects to their correct classes.

This plethora of metrics or figures of merit, originally intended for the binary case and typically focused on one class, is not always directly applicable to the multi-class problem. To extend the use of some metrics to a framework with no single-class emphasis, different methods have been suggested, even to consider the importance of classes in terms of domain experts, scarcity (minority classes), misclassification costs or multiple criteria [27,28]. Thus, performance measures for *K*-class classifiers are still an everlasting issue in literature, notably those with very imbalanced class distributions and/or small datasets [29].

Performance criteria are used, not only to evaluate/validate the constructed models but also to choose between models, to estimate parameters, or to select model components [20,30], situations, among others, where it is useful to represent the global classification performance with a single number [26]. In this context, the present work proposes a modified entropy-based index, a figure of merit that encompasses some of the mentioned figures of merit and that is more sensitive to the variations in the different class-models than similar published indexes.

The following Section 2 summarizes some common figures of merit and introduces the new index, Diagonal Modified Confusion Entropy (DMCEN), together with the basic definitions for the comparisons, whereas Section 3 sheds light on how DMCEN operates along with its ability to detect differences in both sensitivity and specificity.

2. Theory and proposal

In the present work, *K* categories or classes are jointly modelled (compliant class-modelling approaches with the distinction made in Ref. [19]). Therefore, in the following, a *K-class-model* will refer to the set of the *K* individual class-models that are jointly computed and validated against each other. To do this, the training set contains objects belonging to the *K* classes under study.

2.1. Notation

Precisely, to model *K* categories $C_1, C_2, \ldots, C_K$, we have a training set with *I* objects, I_j in each class C_j ($\sum_{j=1}^{K} I_j = I$). With the notation in Ref. [30], the *K*-class-model can be summarized in the so-called confusion matrix **N** in Eq. [1], where n_{jm} is the number of objects belonging to class C_j which are inside the class-model built for class C_m.

$$\mathbf{N} = \begin{pmatrix} n_{11} & n_{12} & \ldots & n_{1m} & \ldots & n_{1K} \\ n_{21} & n_{22} & \ldots & n_{2m} & \ldots & n_{2K} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ n_{j1} & n_{j2} & \ldots & n_{jm} & \ldots & n_{jK} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ n_{K1} & n_{K2} & \ldots & n_{Km} & & n_{KK} \end{pmatrix} \quad (1)$$

In a discriminant situation, which is the context of the confusion matrices, for $j = 1, \ldots, K$, is $\sum_{m=1}^{K} n_{jm} = I_j$, that is, the rows in matrix **N** of Eq [1] sum up to the total number of objects in that class, and $\sum_{j,m} n_{jm} = I$.

However, in the class-modelling setting, neither the rows, nor the total sum of the elements in **N** necessarily meet these equalities. In particular, the rows can add up more or less than the number I_j of objects of C_j in the training set because an object can be inside more than one class-model or outside all of them. To clearly distinguish the different situation when using class-modelling techniques, we will call **N** a model matrix. Throughout the present work, **N** will always denote a model matrix.

From model matrix **N**, we compute the frequency matrix $\mathbf{F} = (f_{jm})$ in equation [2] that contains the rates n_{jm}/I_j. As with Eq. [1], the sum of rates in each row of **F** is not necessarily one, which would be the case for a usual confusion matrix computed with a discriminant method.

$$\mathbf{F}=\begin{pmatrix} f_{11} & f_{12} & \dots & f_{1m} & \dots & f_{1K} \\ f_{21} & f_{22} & \dots & f_{2m} & \dots & f_{2K} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ f_{j1} & f_{j2} & \dots & f_{jj} & \dots & f_{jK} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ f_{K1} & f_{K2} & \dots & f_{Km} & \dots & f_{KK} \end{pmatrix}=\begin{pmatrix} n_{11}/I_1 & n_{12}/I_1 & \dots & n_{1m}/I_1 & \dots & n_{1K}/I_1 \\ n_{21}/I_2 & n_{22}/I_2 & \dots & n_{2m}/I_2 & \dots & n_{2K}/I_2 \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ n_{j1}/I_j & n_{j2}/I_j & \dots & n_{jj}/I_j & \dots & n_{jK}/I_j \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ n_{K1}/I_K & n_{K2}/I_K & \dots & n_{Km}/I_K & \dots & n_{KK}/I_K \end{pmatrix} \tag{2}$$

Irrespective of the type of class-models built, it is not infrequent to define an 'a posteriori' decision rule to assign an object to one of the classes. These decisions with K categories can also be seen as a family of $K(K-1)$ hypothesis tests, $K-1$ for each of the K classes C_j. For each $j = 1, \dots, K$, the null hypothesis H_0 is the same with different alternative hypothesis H_1:

H_0:object **x** belongs to class-model C_j.

H_1:object **x** belongs to class-model C_m, $m = 1, \dots, K$, $m \neq j$.

Symbolically, matrix TEST in Eq. [3] summarizes the different hypothesis tests, in columns the alternative hypothesis of the K-1 tests made with the j-th class in rows as null hypothesis.

$$\mathbf{TEST}=\begin{pmatrix} H_0: C_1 & H_1: C_1 & \dots & H_1: C_1 & \dots & H_1: C_1 & \dots & H_1: C_1 \\ H_1: C_2 & H_0: C_2 & \dots & H_1: C_2 & \dots & H_1: C_2 & \dots & H_1: C_2 \\ \vdots & \vdots & \ddots & \vdots & & \vdots & & \vdots \\ H_1: C_m & H_1: C_m & \dots & H_0: C_m & \dots & H_1: C_m & \dots & H_1: C_m \\ \vdots & \vdots & & \vdots & \ddots & \vdots & & \vdots \\ H_1: C_j & H_1: C_j & \dots & H_1: C_j & \dots & H_0: C_j & \dots & H_1: C_j \\ \vdots & \vdots & & \vdots & & \vdots & \ddots & \vdots \\ H_1: C_K & H_1: C_K & \dots & H_1: C_K & \dots & H_1: C_K & \dots & H_0: C_K \end{pmatrix} \tag{3}$$

With this notation, $\alpha_{jj} = \alpha_j$ is the probability of type I error and, therefore, $1 - \alpha_j$ (probability of correctly assigning an object of C_j to the class-model of C_j) is estimated by f_{jj}, the diagonal terms in matrix **F** of Eq. [2]. Consequently, the sensitivity of the class-model built for C_j is estimated as sens(j) = f_{jj}.

On the other hand, for $m \neq j$, specificity flaws for the class-model of category C_j are in f_{mj} (notice the order of subindexes). In the hypothesis tests context, β_{mj} is the probability of type II error when the alternative hypothesis is the one related to C_m, that is, the probability of (wrongly) accepting in the class-model of C_j an object of class C_m. This probability is thus estimated by f_{mj} and, therefore, $1 - f_{mj}$ estimates the specificity of the class model of C_j as against the class C_m, spec(j, m).

With this notation, matrix **F** is transformed into matrix **S** of sensitivities and specificities in Eq. [4].

$$\mathbf{S}=\begin{pmatrix} s_{11} & s_{12} & \dots & s_{1m} & \dots & s_{1K} \\ s_{21} & s_{22} & \dots & s_{2m} & \dots & s_{2K} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ s_{j1} & s_{j2} & \dots & s_{jj} & \dots & s_{jK} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ s_{K1} & s_{K2} & \dots & s_{Km} & \dots & s_{KK} \end{pmatrix}=\begin{pmatrix} f_{11} & 1-f_{12} & \dots & 1-f_{1m} & \dots & 1-f_{1K} \\ 1-f_{21} & f_{22} & \dots & 1-f_{2m} & \dots & 1-f_{2K} \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ 1-f_{j1} & 1-f_{j2} & \dots & f_{jj} & \dots & 1-f_{jK} \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ 1-f_{K1} & 1-f_{K2} & \dots & 1-f_{Km} & \dots & f_{KK} \end{pmatrix}$$
$$=\begin{pmatrix} \text{sens}(1) & \text{spec}(2,1) & \dots & \text{spec}(m,1) & \dots & \text{spec}(K,1) \\ \text{spec}(1,2) & \text{sens}(2) & \dots & \text{spec}(m,2) & \dots & \text{spec}(K,2) \\ \vdots & \vdots & \ddots & \vdots & & \vdots \\ \text{spec}(1,j) & \text{spec}(2,j) & \dots & \text{sens}(j) & \dots & \text{spec}(K,j) \\ \vdots & \vdots & & \vdots & \ddots & \vdots \\ \text{spec}(1,K) & \text{spec}(2,K) & \dots & \text{spec}(m,K) & \dots & \text{sens}(K) \end{pmatrix} \tag{4}$$

Notice that, like in the remaining matrices, in s_{jm} the first subindex refers to the real class (C_j) of an object and the second to whether it belongs to the class-model constructed for C_m, so the different specificities for the individual class-models appear in the columns of matrix **S** according to hypothesis tests of Eq. [3].

Finally, consideration of the relative size of the classes in the performance of the K-class-model is implicit when moving from **N** to **F**. If a class consists of 10 objects and another of 100, a single wrong allocation in the K-class-model for these classes would imply a decrease of 1/10 or 1/100 in the sensitivity of the corresponding class-model. However, once **F** is estimated, the effect of the size of each class disappears.

2.2. Figures of merit in chemometrics

As mentioned in the introduction, a vast number of performance metrics in machine learning has been found in the literature. Since the present paper focuses on class-modelling in the field of chemometrics, figures of merit (FoMs) based on sensitivity and specificity have been chosen.

To summarize, we have I objects from K categories of interest, $I = I_1 + I_2 + \dots + I_K$, that are used to compute a K-class-model, i.e., a set of K individual class-models. In the following, some of the definitions are repeated to make it easier to refer to them. Thus, for $j = 1, \dots, K$:

The sensitivity of the class-model built for C_j in the K-class-model, the diagonal terms in matrix **S**, Eq. [4], named CSNS(j) in Ref. [30], is commonly defined by

$$s_{jj}=\frac{n_{jj}}{I_j}=1-\alpha_j=\text{CSNS}(j) \tag{5}$$

The specificity of the class-model of C_j to class C_m ($m = 1, 2, \dots, K$, $m \neq j$) is:

$$s_{mj}=1-\frac{n_{mj}}{I_m}=1-\beta_{mj} \tag{6}$$

In ref. [30], in the context of one-class classifiers, from a confusion matrix like **N,** the class specificity, CSPS, is defined globally as against all other classes as "the rate of objects from other classes (not C_j), which are correctly attributed as inconsistent with the target class C_j ", namely:

$$\text{CSPS}(j) = 1-\frac{\sum_{m=1,m\neq j}^{K} n_{mj}}{I-I_j} \tag{7}$$

The same authors also defined an overall quality of classification for the class-model for C_j in terms of the class efficiency, CEFF (called G-mean in Ref. [24]):

$$\mathrm{CEFF}(j) = \sqrt{\mathrm{CSNS}(j) \times \mathrm{CSPS}(j)} \tag{8}$$

Finally, to characterize the entire K-class-model, the total sensitivity, total specificity, and total efficiency are defined [30] as:

$$\mathrm{TSNS} = \frac{\sum_{j=1}^{K} n_{jj}}{I} \tag{9}$$

$$\mathrm{TSPS} = 1 - \frac{\sum_{j=1}^{K} \sum_{m=1, m\neq j}^{K} n_{jm}}{I} \tag{10}$$

$$\mathrm{TEFF} = \sqrt{\mathrm{TSNS} \times \mathrm{TSPS}} \tag{11}$$

Notice that the definition in Eq. [10] implicitly assumes that the sum of the off-diagonal elements n_{jm}, $j \neq m$, is at most I (the total number of objects), which is true for purely confusion matrices but might not be the case for every model matrix $\mathbf{N}$ when $K > 2$. Therefore, for the K-class--model with more than two classes, equations [10,11] should be modified to account for the possibility that an object of class C_j belongs to any (or to all in the worst case) of the $K - 1$ remaining class-models constructed for class C_m ($m \neq j$). Then, a modified total specificity, MTSPS, that applies for $K > 2$, is defined in Eq. [12], and the new overall, modified, total efficiency is in Eq. [13].

$$\mathrm{MTSPS} = 1 - \frac{\sum_{j=1}^{K} \sum_{m=1, m\neq j}^{K} n_{jm}}{(K-1)\, I} \tag{12}$$

$$\mathrm{MTEFF} = \sqrt{\mathrm{TSNS} \times \mathrm{MTSPS}} \tag{13}$$

Due to the greater denominator, MTSPS > TSPS and, contrary to TSPS for model matrices, $0 \leq \mathrm{MTSPS} \leq 1$.

The same observation and correction have been recently published, independently, in Ref. [31] but without modifying the name of the FoMs. In the present work, we will maintain the different notations, MTSPS and MTEFF, to avoid confusion with some previous works that use TSPS and TEFF.

In practice, the figures of merit (FoMs) in Eqs. [9–13] are not sensitive enough to changes in matrix $\mathbf{S}$ (or in model matrix $\mathbf{N}$) resulting in a limited usefulness when their intended use is, for example, to compare different methods (say SIMCA and QDA) in a given problem, or to select the metaparameters of a K-class-model, such as the confidence level of each class when using QDA, for instance.

To illustrate the assertion, let us suppose that we model two categories from a dataset with $I_1 = I_2 = 100$ objects per class ($I = 200$). We have computed two different 2-class-models with model matrices $\mathbf{N_1} = \begin{pmatrix} 100 & 70 \\ 50 & 100 \end{pmatrix}$ and $\mathbf{N}_2 = \begin{pmatrix} 90 & 90 \\ 10 & 70 \end{pmatrix}$. With the definitions in Eqs. [9–11], TSNS = 1, TSPS = 0.4 for $\mathbf{N}_1$, and TSNS = 0.8, TSPS = 0.5 for $\mathbf{N}_2$ whereas, despite their clear differences, both have the same total efficiency $\mathrm{TEFF} = 0.6325 = \sqrt{1 \times 0.4} = \sqrt{0.8 \times 0.5}$.

Of course, both 2-class-models with model matrices $\mathbf{N}_1$ and $\mathbf{N}_2$ are useless from a practical point of view where they will be immediately discarded. However, we are looking for performance criteria that help in conducting a systematic (probably 'blind') selection of the class-models, so the figures of merit should also be sensitive to these situations.

In any case, the insensitivity to the distribution of n_{jm} (and consequently to the one of sensitivities and specificities) illustrated in the previous example worsens as the number of classes increases. When inspecting Eqs. [9–12] it is clear that the FoMs only depend upon some sums and products of values n_{jm} reason why they are the same provided the sums are kept constant.

Other computations of FoMs are proposed [24] for K-class-models by considering K binary situations, namely each C_j against all other ($\cup_{i\neq j} C_i$, that act as the alternative hypothesis of a joint hypothesis test). The definitions of sensitivity and specificity in Ref. [24] are the same as in Eqs. [5,7] but instead of total sensitivity and specificity in Eqs. [9,10], the overall evaluation of the K-class-model is made in terms of *pooled* sensitivity (p-SENS) and *pooled* specificity (p-SPEC), computed as a convex combination of individual sensitivities and specificities, that is:

$$\mathrm{p-SENS} = \sum_{j=1}^{K} w_j \mathrm{CSNS}(j) \tag{14}$$

$$\mathrm{p-SPEC} = \sum_{j=1}^{K} w_j \mathrm{CSPS}(j) \tag{15}$$

with $0 \leq w_j \leq 1$ and $\sum_{j=1}^{K} w_j = 1$. In particular, $w_j = 1/K$ is used in Ref. [25].

These indexes use the same elements as the FoMs previously discussed so a similar lack of responsiveness is expected. Hence, for the modelling of K classes, more sensitive FoMs are needed, both for each class-model and for the overall K-class-model.

2.3. MCEN, an entropy-based figure of merit

In classification contexts, some entropy-based performance criteria (figures of merit) are also used, that can be adapted to the situation here. They are based on the idea that a K-class-model, when applied to a set of objects, reduces their disorganization by including them in the model of each class. In this broad sense, the K-class-model decreases the entropy of the set of objects. The development of this idea is found in Refs. [32,33]. The first one proposes, for the first time, a measure of the uncertainty generated by a K-class-model, called Confusion Entropy (CEN), inspired by Shannon's entropy. This measure is enhanced in Ref. [33] by defining the modified confusion entropy, MCEN, which constitutes the base of our proposal. To introduce it, some previous definitions are needed.

For $j, m = 1, \ldots, K$, eqs. [16,17] contain the definition, for $j \neq m$, of the ratio of frequency f_{jm} subject to class C_j or to class C_m, respectively, and Eq. [18] the definition when $j = m$.

$$R^{j}_{jm} = \frac{f_{jm}}{\sum_{k=1}^{K} (f_{jk} + f_{kj}) - f_{jj}} \tag{16}$$

$$R^{m}_{jm} = \frac{f_{jm}}{\sum_{k=1}^{K} (f_{mk} + f_{km}) - f_{mm}} \tag{17}$$

$$R^{j}_{jj} = 0 \tag{18}$$

In Eqs. [16,17] it is understood that if $f_{jm} = 0$, both ratios are set to zero, irrespective of the corresponding denominator.

As it can be observed, Eqs. [16,17] are fractions with a common numerator, the frequency that the K-class-model (wrongly) includes objects of class C_j into the class-model of C_m. The denominator on its part is computed as the sum of the frequencies of all possible allocations and misallocations involving objects of C_j (Eq. (16)) or of C_m (Eq. (17)). The underlying idea when considering both 'ratios' is that when weighing the specificity of class C_j in relation to class C_m, the reference should consider all the decisions involving both classes C_j and C_m, that is, the corresponding sensitivity and all the involved specificities.

If $I_1 = I_2 = \ldots = I_K$, equations [16,17] are estimates of the probabilities that the K-class-model includes an object of C_j inside the class-model of C_m, but taking into account all errors with objects in classes C_j and C_m because including an object of C_j into the class-model of C_m is as bad as including an object of C_m into the class-model constructed for C_j.

From the ratios in Eq. [16,17], the Modified Confusion Entropy, MCEN, related to class C_j ($j = 1, \ldots, K$) is defined by

$$\mathrm{MCEN}(j) = -\sum_{m=1,m\neq j}^{K}\left(R_{jm}^{j}\log_{2(K-1)}\left(R_{jm}^{j}\right) + R_{mj}^{j}\log_{2(K-1)}\left(R_{mj}^{j}\right)\right) \tag{19}$$

where $R_{jm}^{j}\log_{2(K-1)}(R_{jm}^{j}) = 0$ when $R_{jm}^{j} = 0$, and $R_{mj}^{j}\log_{2(K-1)}(R_{mj}^{j}) = 0$ when $R_{mj}^{j} = 0$.

The overall MCEN for the K-class-model is:

$$\mathrm{MCEN} = \sum_{j=1}^{K} R_j\ \mathrm{MCEN}(j) \tag{20}$$

where the coefficients R_j are defined by

$$R_j = \frac{\sum_{k=1}^{K}\left(f_{jk} + f_{kj}\right) - f_{jj}}{2\sum_{k,m=1}^{K} f_{km} - \lambda\sum_{k=1}^{K} f_{kk}} \tag{21}$$

with

$$\lambda = \begin{cases} \dfrac{1}{2} & \text{if } K = 2 \\ 1 & \text{if } K > 2 \end{cases} \tag{22}$$

A conical combination is a linear combination with non-negative coefficients (scalar weights). If the coefficients add up to one, further to be non-negative, then it is known as a convex combination. Clearly, when $K > 2$ ($\lambda = 1$) the sum of the weights R_j in Eq. [21] is one and MCEN in Eq. [20] is a convex combination of the individual MCEN(j) in Eq. [19]. However, the weighted sum in Eq. [20] is a conical combination when $K = 2$ because, despite all R_j being positive, they do not add up to one (except for $f_{11} = f_{22} = 0$ which would imply that the 2-class-model does not include any object inside the right class-model).

MCEN varies between zero and one. MCEN = 0 corresponds to maximum organization induced by the K-class-model, in other words, every object is correctly inside the right class-model (meaning that sens(j) = 1, j = 1, …, K), and only in that one, so spec(j,m) = 1, for j, m = 1, 2, …, K, $j \neq m$. It is the K-class-model with minimum entropy.

On the other extreme, the K-class-model with maximum entropy (the most disorganized) corresponds to MCEN = 1. It would be a K-class-model with each object inside all the class-models except for its own: sens(j) = 0 (j = 1, …, K) and spec(j,m) = 0, j, m = 1,2, …, K, $j \neq m$.

Table 1 shows six matrices **S**, sensitivity/specificity matrices according to Eq. [4], corresponding to six different 4-class-models. The six matrices have the same values of specificity ($s_{jm} = 1$, $j \neq m$ except for $s_{43} = s_{34} = 0.85$). Furthermore, the first four matrices have sensitivity 1 in all but one class-model, that changes until all four have been covered. **S5** and **S6** 'plays' with asymmetric values of sensitivity along the four individual class-models.

Assuming that there are the same objects in each category for computing the efficiencies in Eqs. [8,11], the corresponding columns in Table 2 show the individual values of class efficiency CEFF(j), Eq. [8], and modified confusion entropy MCEN(j), Eq. [19], for j = 1, …, 4, and also total efficiency, TEFF in Eq. [11], and the overall modified confusion entropy, MCEN in Eq. [20].

With the discussion around Eq. [10], we said that for the case here, $K = 4 > 2$, it is better to use MTEFF in Eq. [13] to avoid negative inconsistent values. Nevertheless, as this was not the case with the matrices in Table 1, we computed the original TEFF in Ref. [30] for comparative purposes.

Finally, to help the reader become familiar with the formulas, Annex 1 shows the detailed computation of MCEN for matrix **S1**. The rest of the values in Table 2 are similarly computed.

As we have already pointed out, TEFF is insensitive to the differences, its value 0.9124 is the same for all six matrices, despite the different class-models. The same behavior would be observed if MTEFF were used, although with a slightly greater value (0.93675). The same insensitiveness applies with p-SENS, Eq. [14], and p-SPEC, Eq. [15], whose values computed with $w_i = 1/4$ are 0.9 and 0.86, respectively, for the six matrices in Table 1.

MCEN on the other hand shows some differences: it is the same for **S1**, **S2**, and **S5** (0.1722), different from the one of **S3** and **S4** (0.1575), and also different from 0.1690 for **S6**. In other words, MCEN distinguishes **S1** when compared to **S3** (or **S4**) and **S6**, but not when compared to **S2** or **S5**, nor when comparing **S3** and **S4**.

When looking at **S1** and **S2**, or **S3** and **S4**, we see that, in fact, they do not represent *true* different class-models, only different *names*, that is, **S1** and **S2** are the same if we interchange categories 1 and 2, and **S3** and **S4** differ in what we call category 3 or 4. The situation is different when comparing to **S5** with rather different sensitivities despite having the same specificities.

A similar (mis)behavior is appreciated when looking at the figures of merit of the individual class-models, MCEN(1) = MCEN(2) = 0, the minimum entropy for the class-models of C_1 and C_2 in all six matrices, which makes no sense for the class-model of C_1 in **S1** or the class-model of C_2 in **S2**, both with sensitivity 0.6, much less than 1, with no distinction with **S3** or **S4** where the first two class-models have perfect sensitivity and specificity. In contrast, CEFF accounts for these differences, with CEFF(1) = 0.7746 and CEFF(2) = 1 in **S1** (the opposite in **S2**), and CEFF(1) = CEFF(2) = 1 for **S3** and **S4**.

The conclusion with respect to the overall measures is that TEFF (and MTEFF) is insensitive to some different distribution of values and the modified confusion entropy MCEN is almost unresponsive to differences in sensitivity, that is, to the diagonal elements of **S.** Given that these elements are the same as in matrix **F**, a closer look at Eqs. [16,17] reveals that the diagonal values f_{jj} only influence the ratios by slightly modifying the denominator; in fact, $R_{jj}^{j} = 0$ by definition.

2.4. DMCEN, a new proposal for a more sensitive entropy-based figure of merit

The previous paragraphs show the lack of sensitivity of MCEN to variations in the K-class-model sensitivity. A new modification of MCEN is proposed, the diagonal modified confusion entropy (DMCEN), to correct this behavior in a way that it becomes useful in class-modelling situations, where sensitivity and specificity are both important.

To do it, the diagonal elements of **F** will be separately considered, but taking into account that they are directly related to the sensitivity of the class-model of C_j, not to misallocations like f_{jm} ($j \neq m$). Therefore, the individual in-diagonal modified confusion entropy, $\mathrm{DMCEN_{id}}$ is defined as:

$$\mathrm{DMCEN_{id}}(j) = 1 - f_{jj}\ ,\ j\ =\ 1,\ 2,\ \ldots,\ K \tag{23}$$

and their weighted mean defines the index for the entire K-class-model as:

$$\mathrm{DMCEN_{id}} = \sum_{j=1}^{K}\mu_j \mathrm{DMCEN}_{id}(j) \tag{24}$$

Table 1
Different matrices of sensitivity and specificity, Eq. [4], for 4-class-models.

S1	S2	S3	S4	S5	S6
$\begin{pmatrix} 0.6 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0.85 \\ 1 & 1 & 0.85 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 0.6 & 1 & 1 \\ 1 & 1 & 1 & 0.85 \\ 1 & 1 & 0.85 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 0.6 & 0.85 \\ 1 & 1 & 0.85 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0.85 \\ 1 & 1 & 0.85 & 0.6 \end{pmatrix}$	$\begin{pmatrix} 0.9 & 1 & 1 & 1 \\ 1 & 0.7 & 1 & 1 \\ 1 & 1 & 1 & 0.85 \\ 1 & 1 & 0.85 & 1 \end{pmatrix}$	$\begin{pmatrix} 0.9 & 1 & 1 & 1 \\ 1 & 0.8 & 1 & 1 \\ 1 & 1 & 0.9 & 0.85 \\ 1 & 1 & 0.85 & 1 \end{pmatrix}$

Table 2
Values of different figures of merit for matrices **S1** to **S6** in Table 1. $w = 0.5$ is used for computing DMCEN(j) and DMCEN.

	Individual class efficiency, CEFF (j), Eq. [8]				Total effic. TEFF, Eq. [11]	Individual modified confusion entropy, MCEN(j), Eq. [19]				MCEN, Eq. [20]	Individual diagonal modified confusion entropy, DMCEN(j), Eq. [25]				DMCEN, Eq. [26]
	$j = 1$	$j = 2$	$j = 3$	$j = 4$		$j = 1$	$j = 2$	$j = 3$	$j = 4$		$j = 1$	$j = 2$	$j = 3$	$j = 4$	
S1	0.7746	1.0000	0.9747	0.9747	0.9124	0.0000	0.0000	0.2781	0.2781	0.1722	0.2000	0.0000	0.1391	0.1391	0.2861
S2	1.0000	0.7746	0.9747	0.9747	0.9124	0.0000	0.0000	0.2781	0.2781	0.1722	0.0000	0.2000	0.1391	0.1391	0.2861
S3	1.0000	1.0000	0.7550	0.9747	0.9124	0.0000	0.0000	0.3333	0.2781	0.1575	0.0000	0.0000	0.3367	0.1391	0.2788
S4	1.0000	1.0000	0.9747	0.7550	0.9124	0.0000	0.0000	0.2781	0.3333	0.1575	0.0000	0.0000	0.1391	0.3667	0.2788
S5	0.9487	0.8367	0.9747	0.9747	0.9124	0.0000	0.0000	0.2781	0.2781	0.1722	0.0500	0.1500	0.1391	0.1391	0.2111
S6	0.9487	0.8944	0.9247	0.9747	0.9124	0.0000	0.0000	0.2901	0.2781	0.1690	0.0500	0.1000	0.1951	0.1391	0.1595

with $\mu_j = \dfrac{\mathrm{DMCEN}_{id}(j)}{\sum_{j=1}^{K}\mathrm{DMCEN}_{id}(j)}$

Notice that with this definition, is $\mathrm{DMCEN}_{\mathrm{id}} = \dfrac{\sum_{j=1}^{K}(1-f_{jj})^2}{\sum_{j=1}^{K}(1-f_{jj})}$. However, any other definition of vector $(\mu_1, \ldots, \mu_K)$ can be used in Eq. [24] to weight the sensitivity of each individual class-model as needed in a particular application.

Finally, a convex combination of the two elements makes the indexes more flexible. Therefore, for $0 \le w \le 1$, the individual diagonal modified confusion entropy is defined for $j = 1, 2, \ldots, K$ by:

$$\mathrm{DMCEN}(j) = w\,\mathrm{MCEN}(j) + (1-w)\mathrm{DMCEN}_{id}(j) \tag{25}$$

and the overall diagonal modified confusion entropy is:

$$\mathrm{DMCEN} = w\,\mathrm{MCEN} + (1-w)\,\mathrm{DMCEN}_{\mathrm{id}} \tag{26}$$

Again, w in Eqs. [25,26], which does not necessarily have to take the same value in the two equations, serves to regulate the relative weight, in a given problem, of sensitivity versus specificity (individually or globally).

Like the rest of detailed FoMs, DMCEN varies between zero and one. However, contrary to sensitivity, specificity, or the remaining FoMs defined up to Eq. [15], the best possible configuration (a matrix **S** of ones) has DMCEN = 0 whereas the maximum value one is for a matrix **S** of zeros, which is the worst situation. Moreover, even if for compatibility we used 1 – DMCEN, the values would not be comparable with the remaining FoMs, reason why we use their original meaning, related to reducing the entropy.

By using $w = 0.5$ in Eqs. [25,26], the final columns of Table 2 show that DMCEN(1) is different from DMCEN(2), except for **S3** and **S4** (where the class-models for C_1 and C_2 are identical). As we have already said, **S1** and **S2** has the same overall DMCEN because class C_1 and C_2 are just interchangeable resulting in the same global structure, though the individual behavior is detected by DMCEN(1) and DMCEN(2). The same happens with **S3** and **S4,** where the interchangeable classes are C_3 and C_4.

However, the differences in the entire 4-class-models are seen by DMCEN: in **S3** and **S4** the 'fails' that reduce sensitivity and specificity are all in two of the constructed class-models (those for C_3 and C_4) whereas in **S1** and **S2** the same fails are distributed in three class-models; thus, they have a smaller value of DMCEN (less disorganized).

Similarly, with the same specificities, the different sensitivities in the 4-class-models in matrices **S5** and **S6** are also detected by both the individual DMCEN(j), $j = 1, \ldots, 4$, and the overall DMCEN. Consequently, with this criterion, the best 4-class-model would be the one summarized in **S6**, where the values of sensitivity less than one are more spread among the class-models, similarly to **S5**, but they are the greatest (0.9 and 0.8 as against 0.9, 0.7, or 0.6).

According to the 'organization' introduced by the 4-class-model as measured by DMCEN, the (decreasing) order in sensitivities in the last four matrices is (0.9, 0.8, 0.9, 1), (0.9, 0.7, 1, 1), and both (1, 1, 1, 0.6) or (1, 1, 0.6, 1) for **S6**, **S5**, and **S4** or **S3**, respectively, with DMCEN equal to 0.1595, 0.2111, and 0.2788, respectively. It is noticeable that this order is not the same as if only sensitivity values were considered. For example, the disorganization of the sensitivities measured directly by Shanon's entropy would be 0.5311, 0.4970, and 0.4422, for **S4**, **S5**, and **S6,** respectively, or 0.2000, 0.1414, and 0.086, respectively, if it was measured by the standard deviation. That means that DMCEN is jointly qualifying the discrepancy in sensitivities and specificities.

Table 3
Benchmark values of DMCEN for different number of classes, K.

K	DMCEN benchmark
2	0.7028
3	0.7144
4	0.7154
5	0.7196
6	0.7234
7	0.7264
8	0.7289
9	0.7309
10	0.7325
11	0.7340
12	0.7351
13	0.7362
14	0.7371
15	0.7378
16	0.7385
17	0.7392
18	0.7397
19	0.7402
20	0.7407

2.5. DMCEN benchmark value for random classification

Ballabio *et al.* [26] introduced the concept of benchmark threshold as the initial criterion to accept or reject a K-class-model on the basis of its performance. It is based on the idea that a K-class-model can be considered informative if it performs better than a random one. To estimate it, the results of a given K-class-model are compared with those obtained by a random K-class-model, which will be the one whose matrix **F** (and **S)** has all the elements equal to 0.5, because DMCEN is computed from frequencies. The benchmark threshold value would then be the DMCEN that corresponds to this random class-modelling.

Table 3 shows some benchmark values of DMCEN for several values of K (from 2 to 20) and section 3.3 shows some additional analyses with $K = 4$ that will help in understanding and clarifying the usefulness of such a benchmark value to assess the K-class-model quality.

2.6. Comparison between DMECEN and MTEFF

The last paragraphs of section 2.4 show some examples where DMCEN is more sensitive than MTEFF. To systematically analyze the behavior of the two performance criteria, we will follow the definitions of consistency and discriminancy in Ref. [34] to compare two arbitrary single-number evaluation measures.

The adaptation of these definitions to compare DMCEN in Eq. [26] and MTEFF in Eq. [13] for a *K*-class model is described in the following.

Let Ψ denote a set of sensitivity/specificity matrices **S**, Eq. [4], computed with different *K*-class-models built with the same dataset. For any two matrices $\mathbf{S}_1$ and $\mathbf{S}_2$ of Ψ, we can compute both FoMs and count the number of times they agree or disagree when evaluating the performance of the two *K*-class-models related to $\mathbf{S}_1$ and $\mathbf{S}_2$. Formally, we define the sets *R* and *T* in Ψ × Ψ (the Cartesian product) by

$$R = \{(\mathbf{S}_1, \mathbf{S}_2) \in \Psi \times \Psi |\ \mathrm{DMCEN}(\mathbf{S}_1) > \mathrm{DMCEN}(\mathbf{S}_2),\ 1 - \mathrm{MTEFF}(\mathbf{S}_1) > 1 - \mathrm{MTEFF}(\mathbf{S}_2)\} \tag{27}$$

$$T = \{(\mathbf{S}_1, \mathbf{S}_2) \in \Psi \times \Psi |\ \mathrm{DMCEN}(\mathbf{S}_1) > \mathrm{DMCEN}(\mathbf{S}_2),\ 1 - \mathrm{MTEFF}(\mathbf{S}_1) < 1 - \mathrm{MTEFF}(\mathbf{S}_2)\} \tag{28}$$

Remember that both DMCEN and MTEFF vary in [0, 1] but they have opposite interpretation: low values of DMCEN indicates better performance (the closer to zero, the better), whereas the best performance with MTEFF corresponds to values closer to one. Therefore, the set *R* in Eq. [27] contains the pairs of matrices for which both FoMs agree in qualifying $\mathbf{S}_2$ as better than $\mathbf{S}_1$, whereas *T* in Eq. [28] contains the pairs of matrices where the FoMs do not agree: $\mathbf{S}_2$ is better than $\mathbf{S}_1$ with DMCEN while $\mathbf{S}_1$ is better than $\mathbf{S}_2$ with MTEFF.

The degree of consistency, *C*, of DMCEN and MTEFF is

$$C = \frac{card(R)}{card(R) + card(T)} \tag{29}$$

where *card* denotes the cardinal number, that is, the number of elements of the corresponding set, *R* or *T*.

Analogously, the following equations [30,31] define subsets *P* and *Q* in Ψ × Ψ that contain the pairs of matrices indistinguishable with MTEFF but not with DMCEN, and those equal with DMCEN and different with MTEFF, respectively.

$$P = \{(\mathbf{S}_1, \mathbf{S}_2) \mid \mathbf{S}_1, \mathbf{S}_2 \in \Psi,\ \mathrm{DMCEN}(\mathbf{S}_1) > \mathrm{DMCEN}(\mathbf{S}_2),\ 1 - \mathrm{MTEFF}(\mathbf{S}_1) = 1 - \mathrm{MTEFF}(\mathbf{S}_2)\} \tag{30}$$

$$Q = \{(\mathbf{S}_1, \mathbf{S}_2) \mid \mathbf{S}_1, \mathbf{S}_2 \in \Psi,\ \mathrm{DMCEN}(\mathbf{S}_1) = \mathrm{DMCEN}(\mathbf{S}_2),\ 1 - \mathrm{MTEFF}(\mathbf{S}_1) > 1 - \mathrm{MTEFF}(\mathbf{S}_2)\} \tag{31}$$

The degree of discriminancy, *D*, for DMCEN over MTEFF is the quotient of the number of elements in *P* and the number of elements in *Q*:

$$D = \frac{card(P)}{card(Q)} \tag{32}$$

For two matrices $\mathbf{S}_1$ and $\mathbf{S}_2$, with $\mathbf{S}_1$ better than $\mathbf{S}_2$ according to DMCEN, a value *C* for the degree of consistency between the FoMs can be seen as the probability that $\mathbf{S}_1$ is better than $\mathbf{S}_2$ also with MTEFF, or vice versa.

On the other hand, if *D* is the degree of discriminancy of DMCEN over MTEFF, the interpretation is that it is *D* times more probable that DMCEN detects a difference between $\mathbf{S}_1$ and $\mathbf{S}_2$ when MTEFF does not.

Clearly, both $C > 0.5$ and $D > 1$ are required to conclude that DMCEN is a better performance criterion than MTEFF. It is worth mentioning that the comparison is made in terms of consistency and discriminancy by 'counting' the decisions made with both FoMs and not by comparing the closeness of their values to any given target value (for instance, MTEFF close to one or DMCEN close to zero), which makes no sense in this case because the values of both FoMs are not comparable, despite varying in the same range.

Table 4
Three different schemas with symmetric sensitivity/specificity matrices for 4-class-models, $0 \le s \le 1$.

SA	SB	SC
$\begin{pmatrix} s & s & s & s \\ s & s & s & s \\ s & s & s & s \\ s & s & s & s \end{pmatrix}$	$\begin{pmatrix} s & 1 & 1 & 1 \\ 1 & s & 1 & 1 \\ 1 & 1 & s & 1 \\ 1 & 1 & 1 & s \end{pmatrix}$	$\begin{pmatrix} 1 & s & s & s \\ s & 1 & s & s \\ s & s & 1 & s \\ s & s & s & 1 \end{pmatrix}$

3. Analysis of the performance of DMCEN

Once the proposed figure of merit has been defined, an analysis of its performance is required. With this goal, several different situations are posed and analyzed. All the cases will be with $K = 4$ categories, which is more than the usual binary situation but with matrices that still can be reasonably handled to illustrate its properties.

3.1. Symmetric matrices with equal values of sensitivity and specificity

With this goal, DMCEN is firstly computed over a series of symmetric sensitivity/specificity matrices detailed in Table 4, all for 4-class-models, that is, for simultaneously handling four different categories.

Different values of sensitivities and specificities *s*, $0 \le s \le 1$, are used with different structure for three types of matrices. The first type of matrix, **SA,** has identical elements, that is, all sensitivities and specificities are set to *s*, with the purpose of relating the value of DMCEN to *s,* the magnitude of sensitivities/specificities of the 4-class-model.

In the second type, matrices **SB**, pair-wise specificities are set to 1, aiming at observing the value of DMCEN as sensitivity *s* of all class-models increases in a scenario of maximum pair-wise specificity.

The third type of matrices **SC** is intended to observe the effect on DMCEN of increasing pair-wise specificities *s* in a framework of class-models with perfect sensitivity.

For $s = 1$, the three matrices coincide in the ideal performance of the *K*-class-model (hence DMCEN = 0). Finally, DMCEN is computed with $w = 0.5$ in all cases and for both Eqs. [25,26].

The computation of DMCEN starts in Eq. [19], whose addends computed with Eq. [16] are the same for all matrices of the same type, but with different values in each type. For example, for **SA**, the frequencies outside the main diagonal are all $1 - s$, so that $R^j_{jm} = \frac{1-s}{6(1-s)+2s-s} = R^j_{mj}$ and thus $\mathrm{MCEN}(j) = -3 \times 2\frac{1-s}{6-5s}\ln\left(\frac{1-s}{6-5s}\right)\frac{1}{\ln(2\times 3)}$, which is also MCEN in Eq. [20] because all the weights R_j in Eq. [21] with $\lambda = 1$ are ¼. Adding the effect of the main diagonal with Eqs. [23,24] is adding $(1-s)$. Thus, with $w = ½$, the value

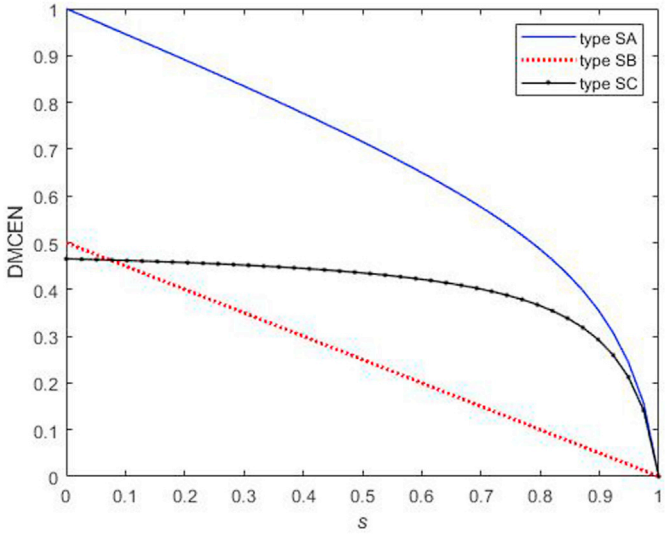

Fig. 1. DMCEN values of the matrices in Table 4 as a function of *s*. Blue line is for **SA**, dotted red line for **SB**, and dash-dotted black line for **SC**. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

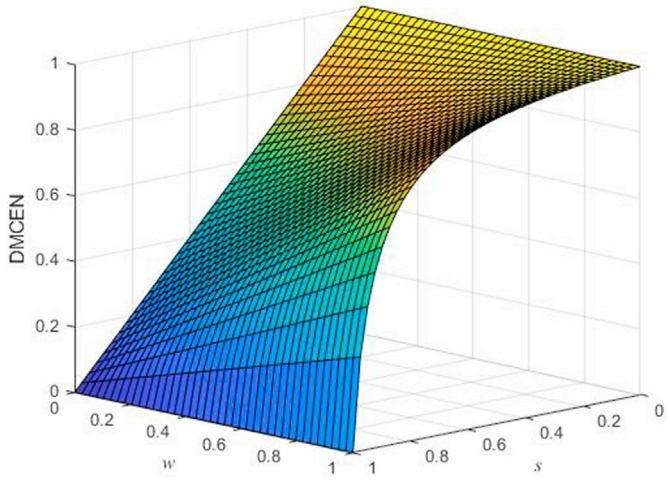

Fig. 2. DMCEN values of matrices SA of Table 4 as a function of s and w.

of DMCEN for matrices **SA** as a function of s is

$$DMCEN(s) = \frac{-3}{\ln(6)} \frac{1-s}{6-5s} \ln\left(\frac{1-s}{6-5s}\right) + \frac{1-s}{2} \tag{33}$$

which is depicted in Fig. 1, blue line. Because DMCEN decreases with the increasing organization of the elements inside the matrix (towards the matrix of ones), the curve follows the expected decreasing behavior: the index steadily decreases when increasing s, that is, when all the sensitivities and specificities increase in the same way. From roughly $s = 0.8$ the decreasing is much more pronounced, reflecting the increasing goodness of the class-models.

For **SB**, the term MCEN is null (all the specificities are perfect) so that DMCEN is only $½(1-s)$ which linearly decreases from 0.5 with slope -0.5, as can be seen in the red dotted line in Fig. 1. Finally, for **SC** the term null is $DMCEN_{id}$ (perfect sensitivity) and, consequently, $DMCEN = ½\,MCEN = ½\,MCEN(j) = \frac{-3}{\ln(6)} \frac{1-s}{7-6s} \ln\left(\frac{1-s}{7-6s}\right)$, which is the black dash-dotted line in Fig. 1. It starts in a better value, reflecting the goodness of sensitivity, but then decreases very slowly until approximately $s = 0.8$, from where it sharply decreases.

The effect of weighting, w, in DMCEN (Eq. (26)) is shown in Fig. 2 on matrices of type **SA.** As s increases, DMCEN decreases. The way of falling depends on the weighting used. When sensitivities are discarded ($w = 1$), DMCEN decreases at a constant rate (as seen in **SB**). On the contrary, with $w = 0$ (specificities are discarded), the rate of decrease is not constant and rises for high s values, as in **SC.**

3.2. Asymmetric K-class-model matrices

In this section, the performance of the proposed FoM, DMCEN, is tested through sensitivity/specificity matrices with varying diagonal elements, and sparse frequency matrices for the off-diagonal elements.

Table 5
Minimum, maximum, and total number of values of the diagonal modified confusion entropy (DMCEN) computed for four different types of sensitivity/specificity matrices in 4-class models (**M1** to **M4**) with the given individual sensitivities.

	Sensitivities of the class-model				DMCEN		
	For C_1	For C_2	For C_3	For C_4	Min	Max	Count
M1	0.90	0.90	0.90	0.90	0.1607	0.1734	11
M2	1.00	1.00	0.80	0.80	0.2097	0.2275	60
M3	1.00	1.00	1.00	0.60	0.3090	0.3281	40
M4	0.60	1.00	1.00	1.00	0.2583	0.2684	2

Four different types of matrices for 4-class-models are defined (**M1** to **M4**) in such a way that the sensitivity of the 4-class-model is kept constant in each type. The kept values are written in Table 5, where it is seen that their sum is always the same but the sensitivities of the individual class-models are not.

Furthermore, for matrices of types **M1** to **M3**, all specificities are one except for three values (0.95, 0.80, and 0.65), which are the same but placed in different off-diagonal elements. In fact, in every type, matrices are generated by allocating these three values to all possible locations within the off-diagonal twelve cells (220 distinct matrices). Additionally, permutations of the three non-unitary specificity values among the three selected allocations must be considered. This results in 1320 (6×220) sensitivity/specificity matrices for each of the three types of matrices (**M1** to **M3**) in Table 5.

The case of **M4** is slightly different because the only non-unitary specificity is 0.4 in (1, 1, 0.4), which is still $0.95 + 0.80 + 0.65 = 1 + 1 + 0.4$. However, only one sensitivity and one specificity are less than one, 0.6 and 0.4, respectively. Consequently, there are only 12 different matrices of type **M4**.

In every matrix from type **M1** to **M4**, both the sum of diagonal elements (trace of the matrix) and the sum of off-diagonal elements are constant, which would be the only information used in the corresponding matrix **N** to compute (assuming the same number of objects per class) total sensitivity (TSNS), total specificity (TSPS), total efficiency (TEFF), modified total efficiency (MTEFF) and also p-SENS and p-SPEC. Consequently, these six FoMs are the same for the $(3 \times 1320) + 12 = 3972$ matrices considered. On the contrary, Table 5 shows the count of the different values of DMCEN obtained in each type of matrices (**M1** to **M4**) along with their maximum and minimum. Indeed, the distinct DMCEN values are in turn obtained, sometimes, in hundreds of matrices.

In any case, the different values of DMCEN when the other FoMs are the same indicate an improvement of the ability of the proposed metric, DMCEN, to distinguish between matrices that the other FoMs do not differentiate.

In the case of matrices of type **M1**, with the same sensitivity in every class-model, the proposed FoM takes 11 different values, which detect the different allocation of the specificity flaws.

Regarding matrices of type **M2**, as differences of specificity between class-models are added to those of sensitivity (only two class-models, those for C_3 and C_4, concentrate the sensitivity flaws), the number of distinct values of DMCEN rises to 60.

For matrices of type **M3**, with a single class-model with sensitivity less than one, DMCEN takes 40 different values, depending on the location of the specificity flaws with respect to the sensitivity value 0.6.

Finally, with the same sensitivity, and a single non-unitary specificity, DMCEN only takes 2 values with matrices of type **M4**. One of them, the maximum 0.2684, occurs in all the matrices where specificity 0.4 and sensitivity 0.6 are in different rows and columns, for example, matrix **SM4max** in Table 6. In contrast, the minimum DMCEN is 0.2583 whenever specificity 0.4 and sensitivity 0.6 are located in the same row or column, as in **SM4min** in Table 6.

Some more examples are shown in Table 6, the already mentioned **SM4min** and **SM4max** that are representative of the minimum and maximum values of DMCEN when using **M4**-type matrices, and **SM1min, SM1max** which takes the minimum and maximum values, 0.1607 and 0.1734, respectively, for matrices of type **M1**.

The worst situation for the matrices of type **M1** is illustrated with **SM1max**, whose overall DMCEN is the maximum value of type **M1** matrices. It is observed how the three non-unitary specificities are located in the same row and column, the first row and first column in the matrix shown. Indeed, class-model of C_1 might accept objects from class C_2, $s_{21} = 0.65$, but also there are objects of C_1 inside the class-models of both C_2, $s_{12} = 0.80$, and C_3, $s_{13} = 0.95$. So specificity problems gather in relation to C_1, related primarily to its own class-model with DMCEN(1) = 0.2514, the largest value among the four class-models, whereas the class-model of C_2 has DMCEN(2) = 0.2220, with 0.20

Table 6
Values of the individual Diagonal Modified Confusion Entropy, DMCEN(j) computed with $w = 0.5$ for four matrices of sensitivities and specificities selected from those in Table 5: **SM1max**: matrix of type **M1** with maximum DMCEN. **SM1min** matrix of type **M1** with minimum DMCEN. **SM4max** matrix of type **M4** with maximum DMCEN. **SM4min**: matrix of type **M4** with minimum DMCEN.

	Matrix	DMCEN (j)			
		$j = 1$	$j = 2$	$j = 3$	$j = 4$
SM1max	$\begin{pmatrix} 0.90 & 0.80 & 0.95 & 1.00 \\ 0.65 & 0.90 & 1.00 & 1.00 \\ 1.00 & 1.00 & 0.90 & 1.00 \\ 1.00 & 1.00 & 1.00 & 0.90 \end{pmatrix}$	0.2514	0.2220	0.0932	0.0500
SM1min	$\begin{pmatrix} 0.90 & 0.65 & 1.00 & 1.00 \\ 1.00 & 0.90 & 1.00 & 1.00 \\ 1.00 & 1.00 & 0.90 & 0.80 \\ 1.00 & 1.00 & 0.95 & 0.90 \end{pmatrix}$	0.1495	0.1495	0.1729	0.1729
SM4max	$\begin{pmatrix} 0.60 & 1.00 & 1.00 & 1.00 \\ 1.00 & 1.00 & 0.40 & 1.00 \\ 1.00 & 1.00 & 1.00 & 1.00 \\ 1.00 & 1.00 & 1.00 & 1.00 \end{pmatrix}$	0.2000	0.1026	0.1026	0.0000
SM4min	$\begin{pmatrix} 0.60 & 1.00 & 1.00 & 1.00 \\ 1.00 & 1.00 & 1.00 & 1.00 \\ 1.00 & 1.00 & 1.00 & 1.00 \\ 0.40 & 1.00 & 1.00 & 1.00 \end{pmatrix}$	0.2967	0.0000	0.0000	0.1026

probability of accepting objects from class C_1 ($s_{12} = 0.80$) together with 0.35 probability that its own objects are inside the class-model of C_1 ($s_{21} = 0.65$). To a lesser extent, the class-model of C_3, with DMCEN(3) = 0.0932, is just affected by a probability 0.05 of accepting objects from class C_1 ($s_{13} = 0.95$). The smallest value, DMCEN(4) = 0.05, is for the class-model of C_4 that just shows a slight sensitivity failure.

Regarding **SM1min,** one of type **M1** matrices with minimum overall DMCEN, specificity flaws are located in different rows and columns, thus affecting three of the class-models. As DMCEN(j)≠0 for all j reveals, all class-models have sensitivity and/or specificity problems. However, they are related with non-empty intersections with only one class. The specificity of the class-models of C_2 is affected only because it contains objects of C_1 (35% as $s_{12} = 0.65$), the one of the class-model of C_3 by accepting objects from only C_4 ($s_{43} = 0.95$) and the class-model of C_4 accepts objects only from C_3 ($s_{34} = 0.80$). The denominators in Eq. [16] are the same for $j = 1,2$ (and for $j = 3,4$) and thus DMCEN(1) equals DMCEN(2). Likewise, DMCEN(3) equals DMCEN(4) though they are slightly greater than DMCEN(j), $j = 1,2$ because the specificity of class-models for C_1 and C_2 suffers for just one failure against another class whereas the class-models for C_3 and C_4 have two failures against another class.

Matrix **SM4max** in Table 6, one of type **M4** matrices with maximum overall DMCEN, has the sensitivity failure in the class-model of C_1, $s_{11} = 0.60$, and thus a non-null DMCEN(1). The single non-unitary specificity in this case is $s_{23} = 0.40$, which corresponds to the class-model of C_3 to class C_2 and thus DMCEN(2) = DMCEN(3) = 0.1026 (both non-null despite the fact that the model of C_2 is perfectly defined), which is a little less than DMCEN(1). The class-model for C_4 is perfectly defined and no objects of C_4 are in any other class-model so DMCEN(4) = 0.

In contrast to **SM4max**, matrix **SM4min** (which is a particular case of the best possible allocation with the **M4** configuration according to DMCEN) has the single non-unitary specificity in $s_{41} = 0.40$, affecting the same class-model with sensitivity $0.6 = s_{11}$. Consequently, DMCEN(1) is the greatest, then 0.1026 for DMCEN(4) because C_4 has objects in the class-model of C_1, and the remaining two class-models, for C_2 and C_3, perfectly defined, thus, with null DMCEN(j), $j = 2,3$.

A similar study but for 4-class-model matrices with equal sensitivities (0.90 in every class-model) has been conducted. The results and discussion are in the supplementary material: Table A1 contains the five types of matrices obtained by varying three non-unitary specificities, along with the count and bounds of the different values obtained for DMCEN. Analogous to Table 6, some particular cases in Table A2 have been analyzed in this situation. Also, a detailed explanation on how to

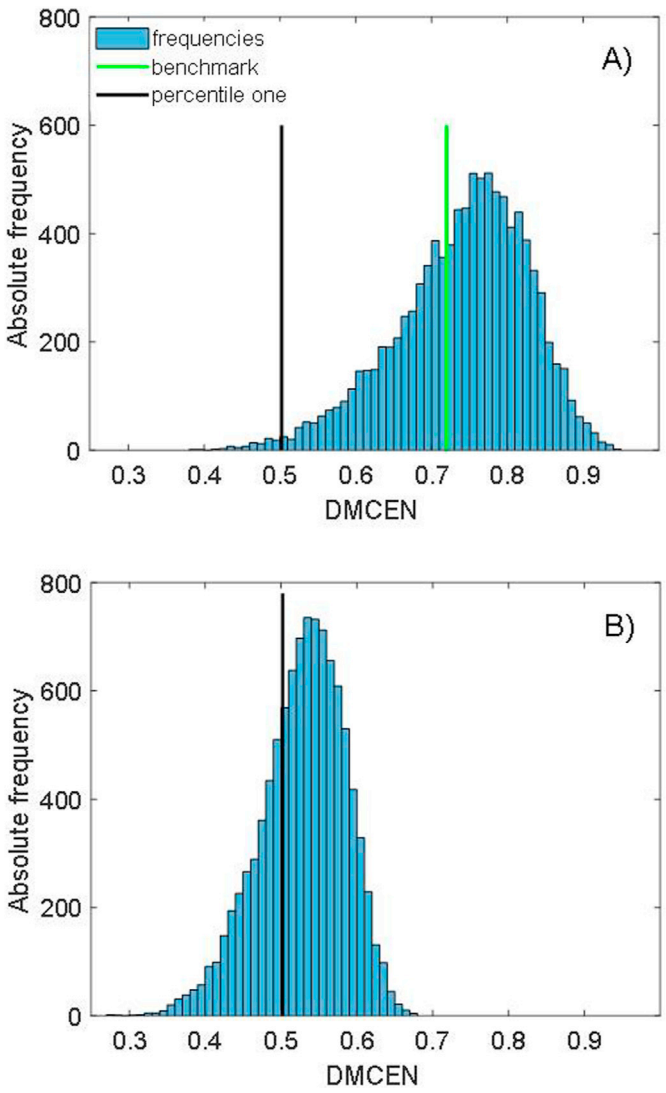

Fig. 3. Histogram of DMCEN values of 10,000 4-class-model matrices of sensitivities and specificities picked uniformly at random from (A) {0, 0.1, 0.2, …, 0.9, 1.0}, (B) {0.5, 0.6, 0.7, 0.8, 0.9, 1}.

interpret the computation and behavior of the individual DMCEN(*j*) is in Tables A3 and A4 of the supplementary material.

As conclusion, when varying specificities with constant sensitivities, Table A1 reveals that, when the difference between the two most extreme specificities increases, overall DMCEN values tend to decrease (notice the decreasing numbers when looking at Table A1 from the first to the last row). On the contrary, when specificities remain constant (in different positions) and sensitivities are changed (Table 5), both the range of DMCEN and its magnitude increase.

3.3. Benchmark value for DMCEN

According to the definitions in section 2.5, a random *K*-class-model has all sensitivities and specificities equal to 0.5. For the particular case of a 4-class-model, that we are using for illustration, that means that, applying Eq. [33] for $s = 0.5$ we have DMCEN = 0.7154, Table 3. Accordingly, any 4-class-model with a value of DMCEN greater than 0.7154 should be directly discarded.

To explore the meaning of this benchmark (threshold) value, we can estimate the distribution of the values of DMCEN, distribution that allows the evaluation of the significance of a particular value of DMCEN obtained for a given 4-class-model.

To illustrate how this works, Fig. 3 depicts histograms of the absolute frequency of 10,000 values of DMCEN obtained in two simulations. The first one is made with 10,000 sensitivity/specificity matrices, whose sixteen elements were randomly picked (with uniform probability) from {0, 0.1, 0.2, …, 0.9, 1}. This covers 4-class-models with very different performance, from very poor to potentially very good. The corresponding histogram is in Fig. 3A), where it is apparent that the distribution of the obtained values of DMCEN is highly asymmetric, the mean is 0.7406, the median 0.7518, and the lower and upper quartiles equal 0.6887 and 0.8031, respectively.

In fact, it seems that there are very few values less than the benchmark 0.7154, which is marked with a green line in Fig. 3A). Precisely, by using the frequencies below this value, we can estimate the probability that DMCEN is less than the benchmark, 0.3454 in this case. Analogously, if we stated, say a 1% significance limit, we can compute the percentile 1 (black line in Fig. 3A)) which is 0.5022. In other words, a 4-class-model with DMCEN less than 0.5022 corresponds to a non-random 4-class-model with 99% confidence level.

To analyze the distribution of the 'suitable' 4-class-models, the histogram in Fig. 3B) corresponds to another 10,000 sensitivity/specificity matrices but whose elements are above the *random* 4-class-model, that is, randomly picked with uniform probability from the reduced set {0.5, 0.6, 0.7, 0.8, 0.9, 1}. As all the sensitivities and specificities are greater than 0.5, the index approaches zero, and the histogram of the values of DMCEN in Fig. 3B) is closer to zero. It is still an asymmetric distribution with mean 0.5282, median of 0.5335, and lower and upper quartiles of 0.4938 and 0.5689, respectively. As expected, all the DMCEN values are less than the benchmark, with a maximum of 0.6733.

Comparatively, the probability of obtaining a DMCEN less than 0.5022 (the previously computed percentile 1 with the histogram in Fig. 3A)) is now 0.30, that is, the percentile 30, marked with a black line in Fig. 3B), that reproduces the idea that in the second case we have discarded all the 'meaningless' 4-class-models, qualified as such according to the benchmark value.

Similar computations can be made for each dataset with *K* classes. It will suffice to re-compute the histogram analogous to the one in Fig. 3B) for the corresponding *K* in Table 3 and, thus, to obtain the probability of having a classifier with a value of DMCEN less than the computed DMCEN.

3.4. Comparison between DMECEN and MTEFF

The previous sections 3.1 and 3.2 describe the behavior of DMCEN throughout some particular cases with four classes, where MTEFF was deliberately kept constant to see the variation of DMCEN. The first conclusion is that DMCEN varies when MTEFF does not so the former performs better than the latter in evaluating *K*-class-models of the illustrated type.

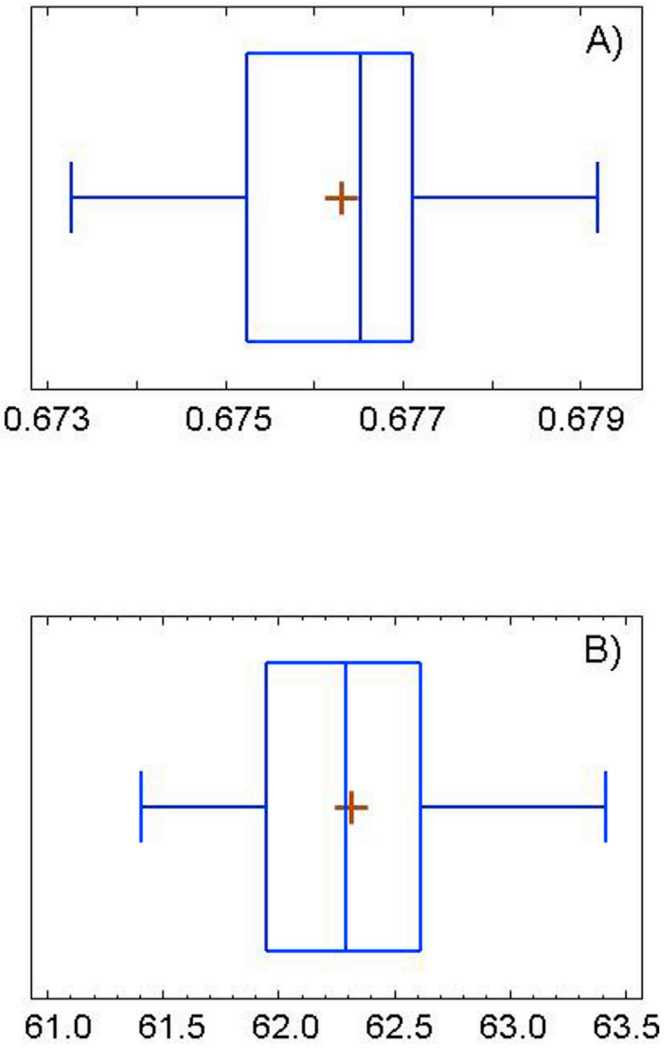

Fig. 4. Box and whisker plot of: (A) the degree of consistency, (B) the degree of discriminancy of DMCEN over MTEFF.

This section is devoted to study the degrees of consistency *C* and discriminancy *D* between both performance measures, according to the definitions in Eqs. [29,32]. With this aim, the set Ψ will be the set containing 100,000 matrices (sensitivity/specificity matrices **S**, Eq [4]), again for four classes. Those **S** matrices are generated by randomly picking the sixteen elements from {0, 0.1, 0.2, …, 0.9,1.0}, with uniform probability.

With the pair-wise comparisons among these 100,000 matrices, *C* and *D* are computed. The procedure is repeated a hundred times. Fig. 4 contains box and whisker plots of the degree of consistency, Fig. 4A), and the degree of discriminancy, Fig. 4B).

Fig. 4A) shows that the 100 values of *C* have a very small dispersion (standard deviation equal to 1.3 10^{-3}) with a high mean and median values, 0.6763 and 0.67653, respectively. That is, for the 4-class-models in Ψ, when DMCEN evaluates one of them better than another, there is 67.6% probability that MTEFF gives the same evaluation.

As for the degree of discriminancy *D*, Fig. 4B) shows that their values vary between 61.41 and 63.42, with almost equal mean and median, 63.31 and 62.29, respectively, and a standard deviation of 0.43. Therefore, it is 62.3 times more probable that DMCEN detects a difference between two 4-class-models in Ψ for which MTEFF does not.

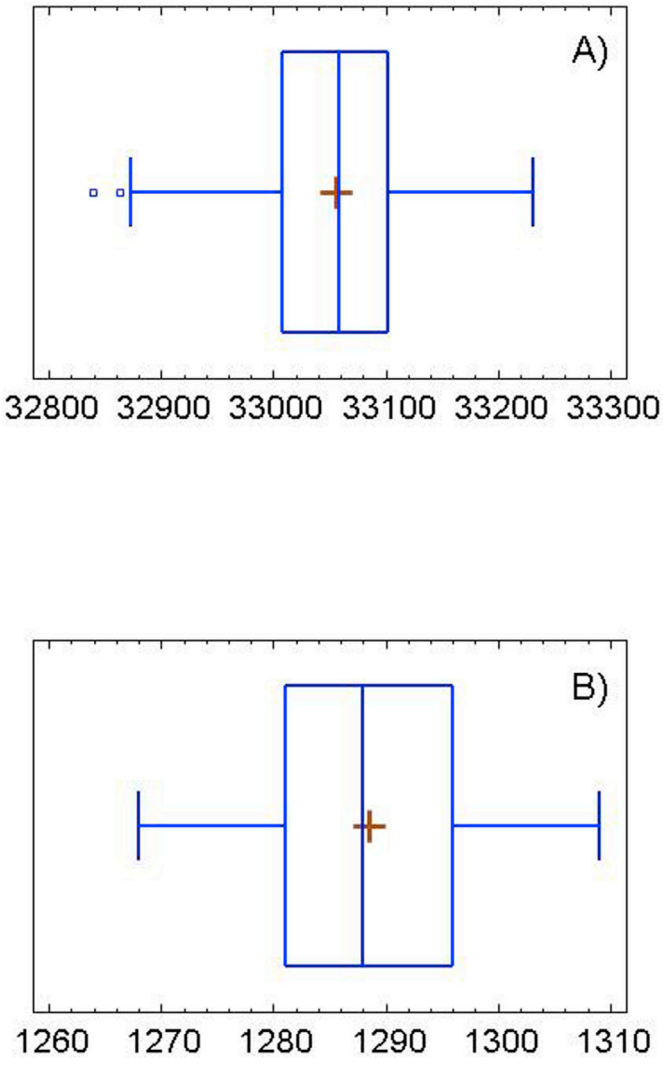

Fig. 5. Box and whisker plot of the number of different values obtained by (A) DMCEN, (B) MTEFF.

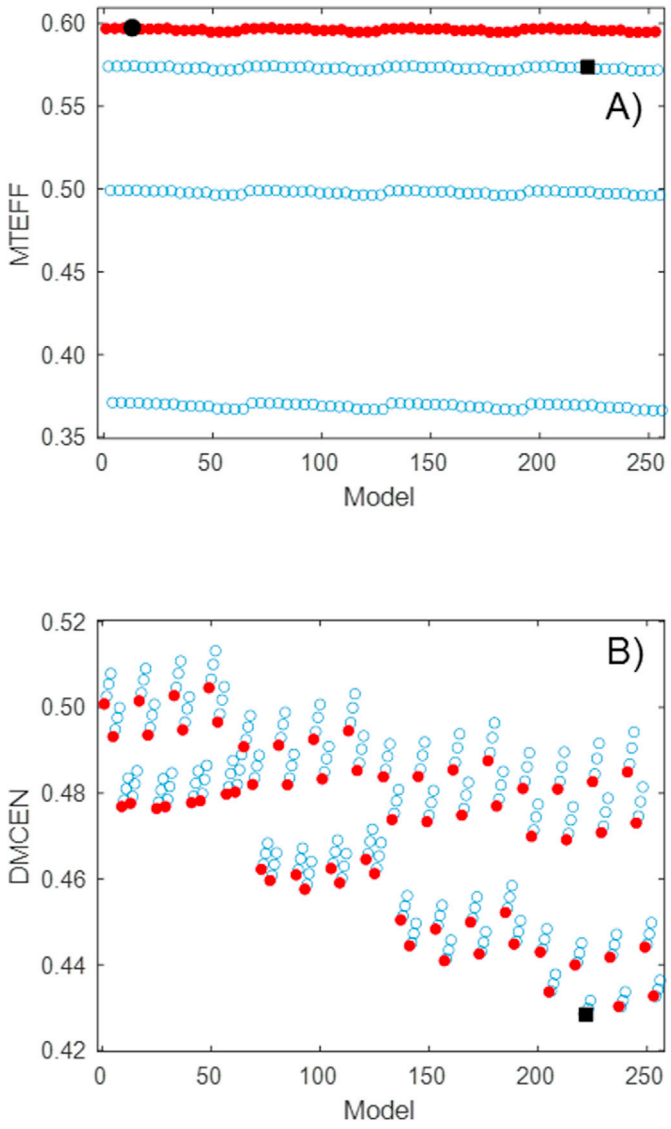

Fig. 6. Values of MTEFF and DMCEN for several different 4-class-models with thyroid data set. A) MTEFF, B) DMCEN with $w = 0.50$. Filled red circles mark the same 4-class-models. The filled black circle, and square are the maximum value of MTEFF, and minimum of DMCEN, respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

Consequently, one can say that DMCEN is a more adequate FoM than MTEFF for the 4-class-models in Ψ.

Besides, the reason of introducing DMCEN is to reduce the insensitivity of MTEFF to some changes in a matrix **S**. To show the improvement, for every 100 sets Ψ, we count the number of different values obtained for both DMCEN (Fig. 5A) and of MTEFF (Fig. 5B). As a result, on average, there are 1,288 different values of MTEFF as against an average of 33,055 of DMCEN, that is, DMCEN is 25.7 times more 'sensitive' than MTEFF when comparing the 100,000 matrices of 4-class-models.

3.5. Illustration of the use of DMCEN with an experimental data set

Unlike the previous sections, the purpose of this section is to show the behavior of DMCEN with experimental data, when varying meta-parameters of a classifier (class-modelling in this case).

To do it, the "allrep" data set from the Thyroid Disease Data Set [35] is considered. It consists of data of 2800 patients distributed in four classes: C_1, replacement therapy; C_2, underreplacement; C_3, over-replacement; and C_4, negative. The five continuous variables have been selected as predictor variables: TSH, thyroid stimulating hormone; T3, triiodothyronine; TT4, total L-thyroxine; T4U, thyroxine uptake; and FTI, free thyroxine index. Objects with some missing value in at least one variable have been removed. In summary, the studied data set has 2,632 patients distributed in 17, 33, 25, and 2,567 patients in each of the four mentioned classes, respectively.

Table 7
Sensitivity/specificity matrices of the 4-class-models selected with minimum DMCEN and maximum MTEFF, in bold.

Confidence level for the class-models of C_1, C_2, C_3, and C_4	MTEFF	DMCEN ($w = 0.50$)	Sensitivity/specificity matrices for C_1, C_2, C_3, and C_4
Model 13 (0.80, 0.80, 0.95, 0.80)	**0.5975**	0.4776	$\begin{pmatrix} 0.71 & 1.00 & 0.28 & 0.78 \\ 0.47 & 0.85 & 0.68 & 0.22 \\ 0.47 & 1.00 & 0.96 & 0.87 \\ 0.41 & 0.15 & 0.60 & 0.91 \end{pmatrix}$
Model 222 (0.95, 0.85, 0.95, 0.85)	0.5739	**0.4285**	$\begin{pmatrix} 0.88 & 1.00 & 0.04 & 0.66 \\ 0.41 & 0.88 & 0.64 & 0.17 \\ 0.47 & 1.00 & 0.96 & 0.87 \\ 0.41 & 0.12 & 0.52 & 0.92 \end{pmatrix}$

UNEQ has been used as a class-modelling method. It consists of building individual hyperellipsoids, at a given confidence level. In that

sense, each class is independently modelled (one-class classifier) but they are jointly evaluated in the form of a sensitivity/specificity matrix of a 4-class-model. Therefore, the metaparameter to be modified is the confidence level at which each hyperellipsoid is built, that is, four confidence levels should be defined.

For each class, confidence levels 0.80, 0.85, 0.90, and 0.95 have been considered, so that 256 different 4-class-models were built. In each of them, MTEFF and DMCEN with $w = 0.5$ in Eq. [26] are evaluated.

In Fig. 6A) the values obtained from MTEF are shown, 4 groups are observed. As the greater MTEFF the better, the group formed by the highest values, marked in red, consists of 64 models whose MTEFF varies from 0.5947 to 0.5975. The maximum (0.5975) is reached in model number 13, the black solid circle in the graph. The characteristics of this model, that is, MTEFF, DMCEN, and the matrix of sensitivities/specificities, are recorded in row 1 of Table 7.

For the same 256 models, the DMCEN values with $w = 0.5$ are shown in Fig. 6B). Clearly, there are much more different values than in Fig. 6A) so that DMCEN discriminates between models better than MTEFF. Besides, the FoM distinguishes between models with different metaparameters, reflecting their structure. For example, models 1 to 4 have confidence level of the first three classes equal to 0.80 and that of the fourth class increasing from 0.80 to 0.95, with the observed corresponding slight increase of DMCEN. Moreover, the red filled circles in Fig. 6B), always at the bottom of the 4-point groups, correspond to the value of DMCEN for the "best" 64 models in Fig. 6A), also in red. The minimum (best value for DMCEN) is obtained on model 222 marked with a solid black square, also in the MTEFF values in Fig. 6A), and its characteristics are in row 2 of Table 7.

Comparing the two rows of Table 7, the class-models differ in the confidence level of all but the third class-model. The consequence is that, in the second row, the sensitivity of all the class-models is improved at the cost of specificity. If, with the problem under study, the researcher wishes to prioritize, say specificity versus sensitivity, then a different w should be defined for computing DMCEN (or even each individual class-model via DMCEN(j)), whereas the values of MTEFF will still be the same.

4. Conclusions

The proposed diagonal modified confusion entropy (DMCEN) as a single overall figure of merit for class-modelling situations with several classes has shown to be more sensitive to the different allocations of sensitivity and specificity of the individual class-models than other usual performance measures for these situations. In particular, the different values of DMCEN, when other usual figures of merit remain constant, indicate an improvement of the ability of the proposed index, DMCEN, to distinguish among class-models that other figures of merit do not differentiate.

A systematic comparison by using the degree of consistency C and the degree of discriminancy D when comparing the proposed DMCEN and the modified total efficiency MTEFF shows that, for a hundred thousand sensitivity/specificity matrices for 4-class-models, C is almost 0.7 on average, well above the needed 0.5, and there is more than 62% probability that DMCEN detects differences when MTEFF does not.

Furthermore, a benchmark threshold value for DMCEN can be computed that allows discarding poor K-class-models that behave worse than a random K-class-model.

The studies conducted show promising behavior of DMCEN to be used as a sole criterion, for example, in a systematic selection of class-models in a given problem, as a response for an experimental design depending on the metaparameters of e.g. SIMCA, or as a fitness function to guide an evolutionary algorithm.

In any case, more studies are probably required for situations where some values of sensitivity and/or specificity are not well estimated, due to class-imbalance or because the class of interest is the least frequent (detection of diseases, bank fraud, etc.).

CRediT authorship contribution statement

O. Valencia: Conceptualization, Formal analysis, Methodology, Writing – original draft, Writing – review & editing. **M.C. Ortiz:** Formal analysis, Conceptualization, Methodology, Supervision, Writing – review & editing, Funding acquisition. **M.S. Sánchez:** Formal analysis, Software, Methodology, Supervision, Writing – original draft, Writing – review & editing. **L.A. Sarabia:** Formal analysis, Conceptualization, Methodology, Software, Supervision, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Authors thank the financial support from Spanish Ministerio de Ciencia, Innovación y Universidades (AEI) and Consejería de Educación de la Junta de Castilla y León through projects CTQ2017-88894-R and BU052P20 respectively (both co-financed with European Regional Development Funds).

Appendix A. Supplementary data

Supplementary data to this article can be found online at https://doi.org/10.1016/j.chemolab.2021.104423.

Annex 1. Computation of MCEN for matrix S1 of Table 1

What follows is the detailed numerical computation of MCEN to better understand why it is equal to zero (perfect classification) when sensitivities are equal to 0.6 in matrix **S1** of Table 1. To make reading easier, the procedure is divided into several steps:

Step 1. To obtain the frequency matrix **F1** from the sensitivity/specificity matrix **S1** by using Eqs. [2,4].

$$\mathbf{S1} = \begin{pmatrix} 0.6 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0.85 \\ 1 & 1 & 0.85 & 1 \end{pmatrix} \mathbf{F1} = \begin{pmatrix} 0.6 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0.15 \\ 0 & 0 & 0.15 & 1 \end{pmatrix}$$

Step 2. Use the formulas in section 2.3 for each class. In most of the cases, since $f_{jm} = 0$, the denominator does not need to be computed. Nevertheless, we show the whole development to make it easier the understanding of the equations.

Class 1: $j = 1$, $m = 2,3,4$ in Eqs. [16,17]		
$R^1_{12} = \frac{f_{12}}{\sum_{k=1}^{4}(f_{1k}+f_{k1})-f_{11}}$	$R^1_{13} = \frac{f_{13}}{\sum_{k=1}^{4}(f_{1k}+f_{k1})-f_{11}}$	$R^1_{14} = \frac{f_{14}}{\sum_{k=1}^{4}(f_{1k}+f_{k1})-f_{11}}$
$R^1_{12} = \frac{0}{1.2-0.6} = 0$	$R^1_{13} = \frac{0}{1.2-0.6} = 0$	$R^1_{14} = \frac{0}{1.2-0.6} = 0$
$R^1_{21} = \frac{f_{21}}{\sum_{k=1}^{4}(f_{1k}+f_{k1})-f_{11}}$	$R^1_{31} = \frac{f_{31}}{\sum_{k=1}^{4}(f_{1k}+f_{k1})-f_{11}}$	$R^1_{41} = \frac{f_{41}}{\sum_{k=1}^{4}(f_{1k}+f_{k1})-f_{11}}$
$R^1_{21} = \frac{0}{1.2-0.6} = 0$	$R^1_{31} = \frac{0}{1.2-0.6} = 0$	$R^1_{41} = \frac{0}{1.2-0.6} = 0$

From Eq. [19], $\mathrm{MCEN}(1) = -\sum_{m=1,m\neq1}^{4}(R^1_{1m}\log_6(R^1_{1m}) + R^1_{m1}\log_6(R^1_{m1}))$where $R^1_{1m}\log_6(R^1_{1m}) = 0$ when $R^1_{1m} = 0$, and $R^1_{m1}\log_6(R^1_{m1}) = 0$ when $R^1_{m1} = 0$.
Therefore, $\mathrm{MCEN}(1) = 0$

Class 2: $j = 2$, $m = 1,3,4$ in Eqs. [16,17].

$R^2_{21} = \frac{f_{21}}{\sum_{k=1}^{4}(f_{2k}+f_{k2})-f_{22}}$	$R^2_{23} = \frac{f_{23}}{\sum_{k=1}^{4}(f_{2k}+f_{k2})-f_{22}}$	$R^2_{24} = \frac{f_{24}}{\sum_{k=1}^{4}(f_{2k}+f_{k2})-f_{22}}$
$R^2_{21} = \frac{0}{2-1} = 0$	$R^2_{23} = \frac{0}{2-1} = 0$	$R^2_{24} = \frac{0}{2-1} = 0$
$R^2_{12} = \frac{f_{12}}{\sum_{k=1}^{4}(f_{2k}+f_{k2})-f_{22}}$	$R^2_{32} = \frac{f_{32}}{\sum_{k=1}^{K}(f_{2k}+f_{k2})-f_{22}}$	$R^2_{42} = \frac{f_{42}}{\sum_{k=1}^{K}(f_{2k}+f_{k2})-f_{22}}$
$R^2_{12} = \frac{0}{2-1} = 0$	$R^2_{32} = \frac{0}{2-1} = 0$	$R^2_{42} = \frac{0}{2-1} = 0$

From eq. [19], $\mathrm{MCEN}(2) = -\sum_{m=1,m\neq1}^{4}(R^2_{2m}\log_6(R^2_{2m}) + R^2_{m2}\log_6(R^2_{m2}))$where $R^2_{2m}\log_6(R^2_{2m}) = 0$ when $R^2_{2m} = 0$ and $R^2_{m2}\log_6(R^2_{m2}) = 0$ when $R^2_{m2} = 0$
$\mathrm{MCEN}(2) = 0$

Class 3: $j = 3$, $m = 1,2,4$ in Eqs. [16,17].

$R^3_{31} = \frac{f_{31}}{\sum_{k=1}^{4}(f_{3k}+f_{k3})-f_{33}}$	$R^3_{32} = \frac{f_{32}}{\sum_{k=1}^{4}(f_{3k}+f_{k3})-f_{33}}$	$R^3_{34} = \frac{f_{34}}{\sum_{k=1}^{4}(f_{3k}+f_{k3})-f_{33}}$
$R^3_{31} = \frac{0}{2.3-1} = 0$	$R^3_{32} = \frac{0}{2.3-1} = 0$	$R^3_{34} = \frac{0.15}{2.3-1} = 0.1154$
$R^3_{13} = \frac{f_{13}}{\sum_{k=1}^{4}(f_{3k}+f_{k3})-f_{33}}$	$R^3_{23} = \frac{f_{23}}{\sum_{k=1}^{K}(f_{3k}+f_{k3})-f_{33}}$	$R^3_{43} = \frac{f_{43}}{\sum_{k=1}^{K}(f_{3k}+f_{k3})-f_{33}}$
$R^3_{13} = \frac{0}{2.3-1} = 0$	$R^3_{23} = \frac{0}{2.3-1} = 0$	$R^3_{43} = \frac{0.15}{2.3-1} = 0.1154$

$\mathrm{MCEN}(3) = -\sum_{m=1,m\neq1}^{4}(R^3_{3m}\log_6(R^3_{3m}) + R^3_{m3}\log_6(R^3_{m3})) = -(0.1154\log_6(0.1154) + 0.1154\log_6(0.1154)) = 0.2781$
Because the remaining R^3_{3m} or R^3_{m3} are zero.
$\mathrm{MCEN}(3) = 0.2781$

Class 4: $j = 4$, $m = 1,2,3$ in Eqs. [16,17].

$R^4_{41} = \frac{f_{41}}{\sum_{k=1}^{4}(f_{4k}+f_{k4})-f_{44}}$	$R^4_{42} = \frac{f_{42}}{\sum_{k=1}^{4}(f_{4k}+f_{k4})-f_{44}}$	$R^4_{43} = \frac{f_{43}}{\sum_{k=1}^{4}(f_{4k}+f_{k4})-f_{44}}$
$R^4_{41} = \frac{0}{2.3-1} = 0$	$R^4_{42} = \frac{0}{2.3-1} = 0$	$R^4_{43} = \frac{0.15}{2.3-1} = 0.1154$
$R^4_{14} = \frac{f_{14}}{\sum_{k=1}^{4}(f_{4k}+f_{k4})-f_{44}}$	$R^4_{24} = \frac{f_{24}}{\sum_{k=1}^{K}(f_{4k}+f_{k4})-f_{44}}$	$R^4_{34} = \frac{f_{34}}{\sum_{k=1}^{K}(f_{4k}+f_{k4})-f_{44}}$
$R^4_{14} = \frac{0}{2.3-1} = 0$	$R^4_{24} = \frac{0}{2.3-1} = 0$	$R^4_{34} = \frac{0.15}{2.3-1} = 0.1154$

$\mathrm{MCEN}(4) = -\sum_{m=1,m\neq1}^{4}(R^4_{4m}\log_6(R^4_{4m}) + R^4_{m4}\log_6(R^4_{m4})) = -(0.1154\log_6(0.1154) + 0.1154\log_6(0.1154)) = 0.2781$
Because the remaining R^4_{4m} or R^4_{m4} are null.
$\mathrm{MCEN}(4) = 0.2781$

Step 3. Compute the coefficients R_i in Eq. [21] of the linear (convex) combination in Eq. [20].

Overall MCEN, Eqs. [20–22]	
$R_1 = \frac{\sum_{k=1}^{4}(f_{1k}+f_{k1})-f_{11}}{2\sum_{k,m=1}^{4}f_{km}-\sum_{k=1}^{4}f_{kk}} = \frac{0.6}{4.2} = 0.1429$	$\mathrm{MCEN}(1) = 0$
$R_2 = \frac{\sum_{k=1}^{4}(f_{2k}+f_{k2})-f_{22}}{2\sum_{k,m=1}^{4}f_{km}-\sum_{k=1}^{4}f_{kk}} = \frac{1}{4.2} = 0.2381$	$\mathrm{MCEN}(2) = 0$
$R_3 = \frac{\sum_{k=1}^{4}(f_{3k}+f_{k3})-f_{33}}{2\sum_{k,m=1}^{4}f_{km}-\sum_{k=1}^{4}f_{kk}} = \frac{1.3}{4.2} = 0.3095$	$\mathrm{MCEN}(3) = 0.2781$
$R_4 = \frac{\sum_{k=1}^{4}(f_{4k}+f_{k4})-f_{44}}{2\sum_{k,m=1}^{4}f_{km}-\sum_{k=1}^{4}f_{kk}} = \frac{1.3}{4.2} = 0.3095$	$\mathrm{MCEN}(4) = 0.2781$
$\mathrm{MCEN} = \sum_{j=1}^{4} R_j\ \mathrm{MCEN}(j) = 0.1722$	

Notice that $0.1429 + 0.2381 + 0.3095 + 0.3095 = 1$ and MCEN is, indeed, a convex combination.

Abbreviations

C	degree of Consistency
CART	Classification And Regression Trees
CEFF	Class Efficiency
CEN	Confusion Entropy
CSNS	Class-model Sensitivity
CSPS	Class-model Specificity
D	degree of Discriminancy
DMCEN	Diagonal Modified Confusion Entropy
$\mathrm{DMCEN}_{\mathrm{id}}$	in-diagonal modified confusion entropy
F	Frequency matrix
FoM	Figure of Merit
LDA	Linear Discriminant Analysis
MCEN	Modified Confusion Entropy
MTEF	Modified Total Efficiency
MTSPS	Modified Total Specificity
N	Model matrix
PLS-DA	Partial Least Squares Discriminant Analysis
p-SENS	pooled Sensitivity
p-SPEC	pooled Specificity
QDA	Quadratic Discriminant Analysis
RDA	Regularized Discriminant Analysis
S	Matrix of sensitivities and specificities
SIMCA	Soft Independent Models of Class Analogy
SVDD	Support Vector Data Description
SVM	Support Vector Machines
TEFF	Total Efficiency
TSNS	Total Sensitivity
TSPS	Total Specificity
UNEQ	Unequal Dispersed Class Models

References

[1] P. Oliveri, C. Malegori, E. Mustorgi, M. Casale, Qualitative pattern recognition in chemistry: theoretical background and practical guidelines, Microchem. J. 162 (2021) 105725, https://doi.org/10.1016/j.microc.2020.105725.

[2] D.F. Morrison, in: Multivariate Statistical Methods, third ed., McGraw-Hill, New York, 1990.

[3] J.H. Friedman, Regularized discriminant analysis, J. Am. Stat. Assoc. 84 (1989) 165–175, https://doi.org/10.1080/01621459.1989.10478752.

[4] M. Barker, W. Rayens, Partial least squares for discrimination, J. Chemometr. 17 (2003) 166–173, https://doi.org/10.1002/cem.785.

[5] L. Breiman, J.H. Friedman, R.A. Olshen, C.J. Stone, Classification and Regression Trees, CRC Press, 1984.

[6] R.G. Brereton, G.R. Lloyd, Support vector machines for classification and regression, Analyst 135 (2010) 230–267, https://doi.org/10.1039/b918972f.

[7] V.N. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[8] M. Sun, A multi-class support vector machine: theory and model, Int. J. Inf. Technol. Decis. Making 12 (2013) 1175–1199, https://doi.org/10.1142/S0219622013500338.

[9] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, J. Platt, Support vector method for novelty detection, in: S.A. Solla, T.K. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems 12, Proceedings of the 1999 Conference, 582–588, MIT Press, 2000.

[10] S. Wold, Pattern recognition by means of disjoint principal components models, Pattern Recogn. 8 (1976) 127–139, https://doi.org/10.1016/0031-3203(76)90014-5.

[11] M.P. Derde, D.L. Massart, UNEQ: a disjoint modelling technique for pattern recognition based on normal distribution, Anal. Chim. Acta 184 (1986) 33–51, https://doi.org/10.1016/S0003-2670(00)86468-5.

[12] D.M. Tax, R.P. Duin, Support vector data description, Mach. Learn. 54 (2004) 45–66, https://doi.org/10.1023/B:MACH.0000008084.60811.49.

[13] M.C. Ortiz, L.A. Sarabia, R. García-Rey, M.D. Luque de Castro, Sensitivity and specificity of PLS-class modelling for five sensory characteristics of dry-cured ham

using visible and near infrared spectroscopy, Anal. Chim. Acta 558 (2006) 125–131, https://doi.org/10.1016/j.aca.2005.11.038.
[14] M.S. Sánchez, M.C. Ortiz, L.A. Sarabia, V. Busto, Class-modelling techniques that optimize the probabilities of false noncompliance and false compliance, Chemometr. Intell. Lab. Syst. 103 (2010) 25–42, https://doi.org/10.1016/j.chemolab.2010.05.007.
[15] M. Forina, P. Oliveri, S. Lanteri, M. Casale, Class-modeling techniques, classic and new, for old and new problems, Chemometr. Intell. Lab. Syst. 93 (2008) 132–148, https://doi.org/10.1016/j.chemolab.2008.05.003.
[16] P. Oliveri, Class-modelling in food analytical chemistry: development, sampling, optimisation and validation issues - a tutorial, Anal. Chim. Acta 982 (2017) 9–19, https://doi.org/10.1016/j.aca.2017.05.013.
[17] R.G. Brereton, One-class classifiers, J. Chemometr. 25 (2011) 225–246, https://doi.org/10.1002/cem.1397.
[18] R.G. Brereton, Pattern recognition in chemometrics, Chemometr. Intell. Lab. Syst. 149 (2015) 90–96, https://doi.org/10.1016/j.chemolab.2015.06.012.
[19] O.Y. Rodionova, P. Oliveri, A.L. Pomerantsev, Rigorous and compliant approaches to one-class classification, Chemometr. Intell. Lab. Syst. 159 (2016) 89–96, https://doi.org/10.1016/j.chemolab.2016.10.002.
[20] D.J. Hand, Evaluating statistical and machine learning supervised classification methods (chapter 3), in: Niall Adams, Edward Cohen (Eds.), Statistical Data Science, World Scientific, 2018, pp. 37–53.
[21] K. Stapor, P. Ksieniewicz, S. García, M. Woźniak, How to design the fair experimental classifier evaluation, Appl. Soft Comput. 104 (2021) 107219, https://doi.org/10.1016/j.asoc.2021.107219.
[22] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recogn. Lett. 30 (2009) 27–38, https://doi.org/10.1016/j.patrec.2008.08.010.
[23] M. Sokolova, G. Lapalme, A systematic analysis of performance measures for classification tasks, Inf. Process. Manag. 45 (2009) 427–437, https://doi.org/10.1016/j.ipm.2009.03.002.
[24] L. Cuadros-Rodríguez, E. Pérez-Castaño, C. Ruiz-Samblás, Quality performance metrics in multivariate classification methods for qualitative analysis, Trends Anal. Chem. 80 (2016) 612–624, https://doi.org/10.1016/j.trac.2016.04.021.
[25] M. Felkin, Comparing classification results between N-array and binary problems, in: F. Guillet, H.J. Hamilton (Eds.), Quality Measures in Data Mining, Springer-Verlag, Berlin, Heidelberg, 2007.
[26] D. Ballabio, F. Grisoni, R. Todeschini, Multivariate comparison of classification performance measures, Chemometr. Intell. Lab. Syst. 174 (2018) 33–44, https://doi.org/10.1016/j.chemolab.2017.12.004.
[27] P. Branco, L. Torgo, R. Ribeiro, Relevance-based evaluation metrics for multi-class imbalanced domains, in: J. Kim, K. Shim, L. Cao, J. Lee, X. Lin, Y. Moon (Eds.), PAKDD 2017, Part I, LNAI 10234, 698–710, 2017, Springer International Publishing AG, 2017, https://doi.org/10.1007/978-3-319-57454-7.
[28] A. Gupta, N. Tatbul, R. Marcus, S. Zhou, I. Lee, J. Gottschlich, Class-weighted evaluation metrics for imbalanced data classification [Preprint 2020], https://arxiv.org/abs/2010.05995.
[29] A. Fernández, S. García, M. Galar, R.C. Prati, B. Krawczyk, F. Herrera, Performance measures, in: Learning from Imbalanced Data Sets, Springer, Cham, 2018, https://doi.org/10.1007/978-3-319-98074-4_3.
[30] A.L. Pomerantsev, O.Y. Rodionova, Multiclass partial least squares discriminant analysis: taking the right way-A critical tutorial, J. Chemometr. 32 (2018), e3030, https://doi.org/10.1002/cem.3030.
[31] A.L. Pomerantsev, O.Y. Rodionova, New trends in qualitative analysis: performance, optimization, and validation of multi-class and soft models, Trends Anal. Chem. 143 (2021) 116372, https://doi.org/10.1016/j.trac.2021.116372.
[32] J.M. Wei, X.Y. Yuan, Q.H. Hu, S.Q. Wang, A novel measure for evaluating classifiers, Expert Syst. Appl. 37 (2010) 3799–3809, https://doi.org/10.1016/j.eswa.2009.11.040.
[33] R. Delgado, J.D. Núñez-González, Enhancing Confusion Entropy (CEN) for binary and multiclass classification, PloS One 14 (2019), e0210264, https://doi.org/10.1371/journal.pone.0210264.
[34] J. Huang, C.X. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Trans. Knowl. Data Eng. 17 (2005) 299–310, https://doi.org/10.1109/TKDE.2005.50.
[35] D. Dua, C. Graff, UCI Machine Learning Repository in, University of California, School of Information and Computer Science, Irvine, CA, 2019. Last visit: 04-09-2021, http://archive.ics.uci.edu/ml.