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 Batuecas s/n, 09001, Burgos, Spain
b Dpt. Química Universidad de Burgos, Facultad de Ciencias, Plaza Misael Batuecas 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
class-models or even outside all of them. Performance criteria, also called figures of merit, relative to class-modelling 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].

Here, [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 \( \text{K-class-model} \) 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 \( C_j \). In equation [1], where \( n_{jm} \) is the number of objects belonging to class \( C_j \) which are inside the class-model built for class \( C_m \).

\[
\begin{pmatrix}
\begin{array}{cccc}
I_{n1} & I_{n2} & \cdots & I_{nk} \\
I_{n1} & I_{n2} & \cdots & I_{nk} \\
\vdots & \vdots & \ddots & \vdots \\
I_{n1} & I_{n2} & \cdots & I_{nk} \\
I_{n1} & I_{n2} & \cdots & I_{nk} \\
I_{n1} & I_{n2} & \cdots & I_{nk} \\
\end{array}
\end{pmatrix}
\]

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

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 \( F = (f_{mn}) \) 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.
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 \( \mathcal{C}_j \). For each \( j = 1, \ldots, K \), the null hypothesis \( H_0 \) is the same with different alternative hypothesis \( H_1 \):

\[
H_0: \text{object } x \text{ belongs to class-model } \mathcal{C}_j.
\]

\[
H_1: \text{object } x \text{ belongs to class-model } \mathcal{C}_m, \ m = 1, \ldots, K, \ m \neq j.
\]

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

\[
\begin{align*}
\text{TEST} = & \begin{pmatrix}
H_0: \mathcal{C}_1, H_1: \mathcal{C}_1 \cdots H_1: \mathcal{C}_1 & \cdots & H_1: \mathcal{C}_1, H_1: \mathcal{C}_1 \\
H_0: \mathcal{C}_2, H_1: \mathcal{C}_2 \cdots H_1: \mathcal{C}_2 & \cdots & H_1: \mathcal{C}_2, H_1: \mathcal{C}_2 \\
: & \cdots & : \\
H_0: \mathcal{C}_n, H_1: \mathcal{C}_m \cdots H_1: \mathcal{C}_m & \cdots & H_1: \mathcal{C}_m, H_1: \mathcal{C}_m
\end{pmatrix}
\end{align*}
\]

With this notation, \( a_j = a_1 \) is the probability of type I error and, therefore, \( 1 - a_j \) (probability of correctly assigning an object of \( \mathcal{C}_j \) to the class-model of \( \mathcal{C}_j \)) is estimated by \( f_{jj} \), the diagonal terms in matrix \( \mathbf{F} \) of Eq. [2]. Consequently, the sensitivity of the class-model built for \( \mathcal{C}_j \) is estimated as \( \text{sens}_j = f_{jj} \).

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

With this notation, matrix \( \mathbf{F} \) is transformed into matrix \( \mathbf{S} \) of sensitivities and specificities in Eq. [4].

\[
\mathbf{S} = \begin{pmatrix}
s_{11} & s_{12} & \cdots & s_{1m} & \cdots & s_{1K} \\
s_{21} & s_{22} & \cdots & s_{2m} & \cdots & s_{2K} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
s_{m1} & s_{m2} & \cdots & s_{mm} & \cdots & s_{mK} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
s_{K1} & s_{K2} & \cdots & s_{Km} & \cdots & s_{KK}
\end{pmatrix}
= \begin{pmatrix}
s_{11} & 1 - f_{11} & \cdots & 1 - f_{1m} & \cdots & 1 - f_{1K} \\
1 - f_{21} & f_{22} & \cdots & 1 - f_{2m} & \cdots & 1 - f_{2K} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 - f_{m1} & 1 - f_{m2} & \cdots & f_{mm} & \cdots & 1 - f_{mK} \\
\vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\
1 - f_{K1} & 1 - f_{K2} & \cdots & 1 - f_{Km} & \cdots & f_{KK}
\end{pmatrix}
\]

\[\text{CSPS}(j) = 1 - \frac{\sum_{m=1,m\neq j}^{K} f_{mj}}{1 - f_{jj}} \quad (7)\]
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]):

\[
\text{CEFF}(j) = \sqrt{\text{CSNS}(j) \times \text{CSPS}(j)}
\]

(8)

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

\[
\text{TSNS} = \frac{\sum_{j=1}^{K} \sum_{m \neq j} n_{jm}}{I}
\]

(9)

\[
\text{MTSPS} = 1 - \frac{\sum_{j=1}^{K} \sum_{m \neq j} n_{jm}}{(K-1)I}
\]

(10)

\[
\text{TEFF} = \sqrt{\text{TSNS} \times \text{MTSPS}}
\]

(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 \( N \) 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).

\[
\text{MTSPS} = 1 - \frac{\sum_{j=1}^{K} \sum_{m \neq j} n_{jm}}{(K-1)I}
\]

(12)

\[
\text{MTEFF} = \sqrt{\text{TSNS} \times \text{MTSPS}}
\]

(13)

Due to the greater denominator, \( \text{MTSPS} \geq \text{TSPS} \) and, contrary to \( \text{TSPS} \) for model matrices, \( 0 \leq \text{MTEFF} \leq 1 \).

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, TSNS 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 \( S \) (or in model matrix \( N \)) 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.e., 200). We have computed two different 2-class-models with model matrices \( N_1 = \begin{bmatrix} 100 & 70 \\ 50 & 100 \end{bmatrix} \) and \( N_2 = \begin{bmatrix} 90 & 90 \\ 10 & 70 \end{bmatrix} \). With the definitions in Eqs. [9–11],

\[
\text{TSNS} = 1, \text{TSPS} = 0.4 \text{ for } N_1, \text{ and } \text{TSNS} = 0.8, \text{TSPS} = 0.5 \text{ for } N_2 \text{ whereas, despite their clear differences, both have the same total efficiency \( \text{TEFF} = 0.6325 = \sqrt{1 \times 0.4} = \sqrt{0.8} \times 0.5}. \]

(14)

Of course, both 2-class-models with model matrices \( N_1 \) and \( N_2 \) 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 (\( \cup_{j \neq j} G \), 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:

\[
p - \text{SENS} = \sum_{j=1}^{K} w_j \text{CSNS}(j)
\]

(14)

\[
p - \text{SPEC} = \sum_{j=1}^{K} w_j \text{CSPS}(j)
\]

(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 \( j = m \).

\[
R_{jm}^e = \frac{f_{jm}}{\sum_{k=1}^{K} (f_{jk} + f_{km}) - f_{jm}}
\]

(16)

\[
R_{jm}^e = \frac{f_{jm}}{\sum_{k=1}^{K} (f_{jk} + f_{km}) - f_{jm}}
\]

(17)

\[
R_{jm} = 0
\]

(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
MCEN\((j) = -\frac{\sum_{\alpha=1}^{K} \left( R_{\alpha j} \log_2(R_{\alpha j}) + R_{\alpha j} \log_2(1-R_{\alpha j}) \right)}{\sum_{\alpha=1}^{K} R_{\alpha j}} \) \tag{19}

where \( R_{\alpha j} \log_2(R_{\alpha j}) \) is equal to 0 when \( R_{\alpha j} = 0 \), and \( R_{\alpha j} \log_2(1-R_{\alpha j}) \) is also equal to 0 when \( R_{\alpha j} = 1 \).

The overall MCEN for the \( K \)-class-model is:

\[
\text{MCEN} = \sum_{j=1}^{K} R_{j} \cdot \text{MCEN}(j) \tag{20}
\]

where the coefficients \( R_{j} \) are defined by

\[
R_{j} = \frac{\sum_{i=1}^{K} (f_{ji} + f_{j}) - f_{j}}{2 \sum_{i=1}^{K} f_{j} - \delta \sum_{i=1}^{K} f_{ik}} \tag{21}
\]

with

\[
\lambda = \begin{cases} 
1 \frac{1}{2} & \text{if } K = 2 \\
1 & \text{if } K > 2
\end{cases}
\]

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 \( K > 2 \) 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\((j) = 1, j = 1, \ldots, K\), and only in that one, so spec\((j,m) = 1, j = 1, m = 1, \ldots, K, j \neq m\). It is the \( K \)-class-model with maximum entropy. On the other hand, the \( K \)-class-model with maximum entropy (the most disorganized) corresponds to MCEN = 1. It would be a \( K \)-class-model with every object inside all the class-models except for its own: sens\((j) = 0 (j = 1, \ldots, K) \) and spec\((j,m) = 0, j, m = 1, \ldots, K, j \neq m\).

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 (\( s_{1j} = 1, j \neq m \) except for \( s_{34} = s_{43} = 0.85 \)). Furthermore, the first four matrices have sensitivity 1 in all but one class-model, that changes until all four have been covered. The remaining six matrices have asymmetric values of sensitivity along the four 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 \( j = 1, \ldots, 4\), 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, \( K = 4 \geq 2 \), 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 insensitivity 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 \( C_0 \) 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, \( R_{jj} = 0 \) by definition.

### 2.4. DMCM, 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 (DMCM), 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 \( f_{mn} (j \neq m) \). Therefore, the individual in-diagonal modified confusion entropy, DMCM\(_{ij}\) is defined as:

\[
\text{DMCM}_{ij} = 1 - f_{jj}, \quad j = 1, 2, \ldots, K
\]

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

\[
\text{DMCM}_{\alpha} = \sum_{j=1}^{K} \mu_j \cdot \text{DMCM}_{ij} \tag{24}
\]
with \( \mu_j = \frac{\text{DMCEN}_{id}(j)}{\sum_{k=1}^{K} \text{DMCEN}_{id}(j)} \)

Notice that with this definition, is DMCEN_{id} = \( \sum_{j=1}^{K} (1-w)^2 \frac{\mu_j}{\sum_{k=1}^{K} (1-w)^2 \mu_k} \). 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 < w < 1, the individual diagonal modified confusion entropy is defined for \( j = 1, 2, \ldots, K \) by:

\[
\text{DMCEN}(j) = w \text{MCEN}(j) + (1-w)\text{DMCEN}_{id}(j)
\]

and the overall diagonal modified confusion entropy is:

\[
\text{DMCEN} = w \text{MCEN} + (1-w) \text{DMCEN}_{id}
\]

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 (individuality 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 C1 and C2 are identical). As we have already said, S1 and S2 has the same overall DMCEN because class C1 and C2 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 C3 and C4.

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 C2 and C3) 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(1), j = 1, ..., 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 Shannon’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.

### 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 P (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 DMCEN 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.

### 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.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>j = 1</td>
<td>j = 2</td>
<td>j = 3</td>
<td>j = 4</td>
<td>j = 1</td>
</tr>
<tr>
<td>0.7746 1.0000 0.9747 0.9747</td>
<td>0.9124</td>
<td>0.0000 0.0000 0.2781 0.2781</td>
<td>0.1722</td>
<td>0.2000 0.0000 0.1391 0.1391</td>
</tr>
<tr>
<td>1.0000 0.7746 0.9747 0.9747</td>
<td>0.9124</td>
<td>0.0000 0.0000 0.2781 0.2781</td>
<td>0.1722</td>
<td>0.2000 0.0000 0.1391 0.1391</td>
</tr>
<tr>
<td>1.0000 1.0000 0.7550 0.9747</td>
<td>0.9124</td>
<td>0.0000 0.0000 0.3333 0.2781</td>
<td>0.1575</td>
<td>0.0000 0.0000 0.3367 0.1391</td>
</tr>
<tr>
<td>1.0000 0.9487 0.8367 0.9747</td>
<td>0.9124</td>
<td>0.0000 0.0000 0.2781 0.2781</td>
<td>0.1722</td>
<td>0.0500 0.1500 0.1391 0.1391</td>
</tr>
<tr>
<td>0.9487 0.8944 0.9247 0.9747</td>
<td>0.9124</td>
<td>0.0000 0.0000 0.2901 0.2781</td>
<td>0.1690</td>
<td>0.0500 0.1000 0.1591 0.1591</td>
</tr>
</tbody>
</table>

### Table 3

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

<table>
<thead>
<tr>
<th>K</th>
<th>DMCEN benchmark</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>0.7028</td>
</tr>
<tr>
<td>3</td>
<td>0.7144</td>
</tr>
<tr>
<td>4</td>
<td>0.7154</td>
</tr>
<tr>
<td>5</td>
<td>0.7196</td>
</tr>
<tr>
<td>6</td>
<td>0.7234</td>
</tr>
<tr>
<td>7</td>
<td>0.7264</td>
</tr>
<tr>
<td>8</td>
<td>0.7289</td>
</tr>
<tr>
<td>9</td>
<td>0.7309</td>
</tr>
<tr>
<td>10</td>
<td>0.7325</td>
</tr>
<tr>
<td>11</td>
<td>0.7340</td>
</tr>
<tr>
<td>12</td>
<td>0.7351</td>
</tr>
<tr>
<td>13</td>
<td>0.7362</td>
</tr>
<tr>
<td>14</td>
<td>0.7371</td>
</tr>
<tr>
<td>15</td>
<td>0.7378</td>
</tr>
<tr>
<td>16</td>
<td>0.7385</td>
</tr>
<tr>
<td>17</td>
<td>0.7392</td>
</tr>
<tr>
<td>18</td>
<td>0.7397</td>
</tr>
<tr>
<td>19</td>
<td>0.7402</td>
</tr>
<tr>
<td>20</td>
<td>0.7407</td>
</tr>
</tbody>
</table>
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 $\Psi$ 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 $S_1$ and $S_2$ of $\Psi$, 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 $S_1$ and $S_2$. Formally, we define the sets $R$ and $T$ in $\Psi \times \Psi$ (the Cartesian product) by

$$R = \{(S_1, S_2) \in \Psi \times \Psi | \text{DMCEN}(S_1) > \text{DMCEN}(S_2), \ 1 - \text{MTEFF}(S_1) > 1 - \text{MTEFF}(S_2)\}$$

$$T = \{(S_1, S_2) \in \Psi \times \Psi | \text{DMCEN}(S_1) > \text{DMCEN}(S_2), \ 1 - \text{MTEFF}(S_1) < 1 - \text{MTEFF}(S_2)\}$$

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 $S_2$ as better than $S_1$, whereas $T$ in Eq. [28] contains the pairs of matrices where the FoMs do not agree: $S_2$ is better than $S_1$ with DMCEN while $S_1$ is better than $S_2$ with MTEFF.

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

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

where $\text{card}$ 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 $\Psi \times \Psi$ that contain the pairs of matrices indistinguishable with MTEFF but not with DMCEN, and those equal with DMCEN and different with MTEFF, respectively.

$$P = \{(S_1, S_2) \in \Psi \times \Psi | \text{DMCEN}(S_1) > \text{DMCEN}(S_2), \ 1 - \text{MTEFF}(S_1) = 1 - \text{MTEFF}(S_2)\}$$

$$Q = \{(S_1, S_2) \in \Psi \times \Psi | \text{DMCEN}(S_1) = \text{DMCEN}(S_2), \ 1 - \text{MTEFF}(S_1) > 1 - \text{MTEFF}(S_2)\}$$

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 $Q$.

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

For two matrices $S_1$ and $S_2$, with $S_1$ better than $S_2$ according to DMCEN, a value $C$ for the degree of consistency between the FoMs can be seen as the probability that $S_1$ is better than $S_2$ 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 $S_1$ and $S_2$ 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.

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 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 $s$, $0 \leq s \leq 1$, 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 $R_{m}^{h} = \frac{1}{4} - \frac{1}{2} \times (1 - s) = R_{m}^{h}$ and thus $MCEN(j) = -3 \times \frac{2^{rac{1}{6}} \ln \left(\frac{1 - s}{s}\right)}{\ln(2) - 3}$, which is also MCEN in Eq. [20] because all the weights $R_{j}$ in Eq. [21] with $\lambda = 1$ are $\frac{1}{6}$. Adding the effect of the main diagonal with Eqs. [23,24] is adding $(1 - s)$. Thus, with $w = \frac{1}{5}$, 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.)](image)
of DMCEN for matrices SA as a function of s and w.

\[
DMCEN(s) = \frac{3}{\ln(6)} \ln\left(\frac{1-s}{6-5s}\right) + \frac{1-s}{2}
\]  

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 \(\frac{3}{\ln(6)}(1-s)\) 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\(_{\text{SA}}\) (perfect sensitivity) and, consequently, DMCEN = \(\frac{1}{2}\) MCEN = \(\frac{1}{2}\) MCEN\((j) = \frac{3}{\ln(6)} \ln\left(\frac{1-j}{6-5j}\right)\), 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 specificities 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>
<thead>
<tr>
<th>S sensitivities of the class-model</th>
<th>DMCEN</th>
<th>Min</th>
<th>Max</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>For C1</td>
<td>For C2</td>
<td>For C3</td>
<td>For C4</td>
<td></td>
</tr>
<tr>
<td>M1</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
<td>0.90</td>
</tr>
<tr>
<td>M2</td>
<td>1.00</td>
<td>1.00</td>
<td>0.80</td>
<td>0.80</td>
</tr>
<tr>
<td>M3</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>0.60</td>
</tr>
<tr>
<td>M4</td>
<td>0.60</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

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 \times 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 specificity 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 C3 and C4, 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 C1 might accept objects from class C2, s12 = 0.65, but also there are objects of C1 inside the class-models of both C2, s12 = 0.80, and C3, s13 = 0.95. So specificity problems gather in relation to C1, 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 C2 has DMCEN(2) = 0.2200, with 0.20
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 \( C_2 \), with 
\[ \text{DMCEN}(3) = 0.0932, \]
is just affected by a probability 0.05 of accepting objects from class \( C_1 \) \((s_{12} = 0.95)\). The smallest value, 
\[ \text{DMCEN}(4) = 0.05, \]
is for the class-model of \( C_4 \) that just shows a slight sensitivity failure.

Regarding \( \text{SM1}\text{min} \), one of type \( \text{M1} \) 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 \( \text{SM4}\text{max} \) in Table 6, one of type \( \text{M4} \) matrices with maximum overall DMCEN, has the sensitivity failure in the class-model of \( C_1 \), \( s_{12} = 0.60\), 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 \( C_2 \) to class \( C_3 \) 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 \( \text{SM4}\text{max} \), matrix \( \text{SM4}\text{min} \) (which is a particular case of the best possible allocation with the \( \text{M4} \) configuration according to DMCEN) has the single non-unitary specificity in \( s_{12} = 0.40\), 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...
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 DMCEN 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.

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. 4(A), and the degree of discriminancy, Fig. 4(B).

Fig. 4A) shows 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.
Consequently, one can say that DMCEN is a more adequate FoM than MTEFF for the 4-class-models in $\Psi$.

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 $\Psi$, 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; $C_3$, overreplacement; and $C_4$, negative. The five continuous variables have been selected as predictor variables: TSH, thyroid stimulating hormone; T3, triiodothyronine; TT4, total $\tau$-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.

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

---

**Table 7**

<table>
<thead>
<tr>
<th>Confidence level for the class-models of $C_1$, $C_2$, $C_3$, and $C_4$</th>
<th>MTEFF</th>
<th>DMCEN $(w = 0.50)$</th>
<th>Sensitivity/specificity matrices for $C_1$, $C_2$, $C_3$, and $C_4$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model 13 (0.80, 0.80, 0.95, 0.80)</td>
<td>0.5975</td>
<td>0.4776</td>
<td>0.71 1.00 0.28 0.78</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.47 0.85 0.68 0.22</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.47 1.00 0.96 0.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.41 0.15 0.60 0.91</td>
</tr>
<tr>
<td>Model 222 (0.95, 0.85, 0.95, 0.85)</td>
<td>0.5739</td>
<td>0.4285</td>
<td>0.88 1.00 0.04 0.66</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.41 0.88 0.64 0.17</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.47 1.00 0.96 0.87</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0.41 0.12 0.52 0.92</td>
</tr>
</tbody>
</table>
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 DMCEFN with \( w = 0.5 \) in Eq. [26] are evaluated.

In Fig. 6A) the values obtained from MTEFF 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, DMCEFN, and the matrix of sensitivities/specificities, are recorded in row 1 of Table 7.

For the same 256 models, the DMCEFN values with \( w = 0.5 \) are shown in Fig. 6B). Clearly, there are much more different values than in Fig. 6A) so that DMCEFN 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 DMCEFN. Moreover, the red filled circles in Fig. 6B), always at the bottom of the 4-point groups, correspond to the value of DMCEFN for the “best” 64 models in Fig. 6A), also in red. The minimum (best value for DMCEFN) 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 DMCEFN (or even each individual class-model via DMCEFN(j)), whereas the values of MTEFF will still be the same.

4. Conclusions

The proposed diagonal modified confusion entropy (DMCEFN) 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 DMCEFN, when other usual figures of merit remain constant, indicate an improvement of the ability of the proposed index, DMCEFN, 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 discrimnancy \( D \) when comparing the proposed DMCEFN 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 DMCEFN detects differences when MTEFF does not.

Furthermore, a benchmark threshold value for DMCEFN 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 DMCEFN 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].

\[
S1 = \begin{pmatrix}
0.6 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 0.85 & 1 \\
1 & 1 & 0.85 & 1
\end{pmatrix}
\]

\[
F1 = \begin{pmatrix}
0.6 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0.15 \\
0 & 0 & 0.15 & 1
\end{pmatrix}
\]
Step 2. Use the formulas in section 2.3 for each class. In most of the cases, since \( f_m = 0 \), the denominator does not need to be computed. Nevertheless, we show the whole development to make it easier the understanding of the equations.

<table>
<thead>
<tr>
<th>Class 1: ( j = 1 ), ( m = 2,3,4 ) in Eqs. [16,17]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{11}^2 = \frac{f_{21}}{\sum_{x=1}^{j} (fa + fa_1) - f_{21}} )</td>
</tr>
<tr>
<td>( R_{14}^2 = \frac{0}{12 - 0.6} )</td>
</tr>
</tbody>
</table>

From Eq. [19], \( \text{MCEN}(1) = -\sum_{m=1}^{4} \log \phi (R_{1m}^2) + R_{1m}^2 \log \phi (R_{1m}^2) \) where \( R_{1m}^2 \log \phi (R_{1m}^2) = 0 \) when \( m = 0 \), and \( R_{1m}^2 \log \phi (R_{1m}^2) = 0 \) when \( R_{1m}^2 = 0 \). Therefore, \( \text{MCEN}(1) = 0 \)

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

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{21}^2 = \frac{f_{21}}{\sum_{x=1}^{j} (fa + fa_2) - f_{21}} )</td>
</tr>
<tr>
<td>( R_{22}^2 = \frac{0}{2 - 4} )</td>
</tr>
</tbody>
</table>

From Eq. [19], \( \text{MCEN}(2) = -\sum_{m=1}^{4} \log \phi (R_{2m}^2) + R_{2m}^2 \log \phi (R_{2m}^2) \) where \( R_{2m}^2 \log \phi (R_{2m}^2) = 0 \) when \( R_{2m}^2 = 0 \) and \( R_{2m}^2 \log \phi (R_{2m}^2) = 0 \) when \( R_{2m}^2 = 0 \). Therefore, \( \text{MCEN}(2) = 0 \)

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

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{31}^2 = \frac{f_{31}}{\sum_{x=1}^{j} (fa + fa_3) - f_{31}} )</td>
</tr>
<tr>
<td>( R_{32}^2 = \frac{0}{2.3 - 1} )</td>
</tr>
</tbody>
</table>

\( \text{MCEN}(3) = -\sum_{m=1}^{4} \log \phi (R_{3m}^2) + R_{3m}^2 \log \phi (R_{3m}^2) \) = \(- (0.1154 \log \phi (0.1154) + 0.1154 \log \phi (0.1154)) = 0.2781 \)

Because the remaining \( R_{3m}^2 \) or \( R_{4m}^2 \) are zero.

\( \text{MCEN}(3) = 0.2781 \)

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

<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( R_{41}^2 = \frac{f_{41}}{\sum_{x=1}^{j} (fa + fa_4) - f_{41}} )</td>
</tr>
<tr>
<td>( R_{42}^2 = \frac{0}{2.3 - 1} )</td>
</tr>
</tbody>
</table>

\( \text{MCEN}(4) = -\sum_{m=1}^{4} \log \phi (R_{4m}^2) + R_{4m}^2 \log \phi (R_{4m}^2) \) = \(- (0.1154 \log \phi (0.1154) + 0.1154 \log \phi (0.1154)) = 0.2781 \)

Because the remaining \( R_{4m}^2 \) or \( R_{4m}^2 \) are null.

\( \text{MCEN}(4) = 0.2781 \)
Step 3. Compute the coefficients $R_i$ in Eq. [21] of the linear (convex) combination in Eq. [20].

<table>
<thead>
<tr>
<th>Overall MCEN, Eqs. [20–22]</th>
</tr>
</thead>
<tbody>
<tr>
<td>$R_1 = \frac{\sum_{k=1}^{P} (f_k + f_a) - f_1}{\sum_{k=1}^{P} f_k - \sum_{k=1}^{P} f_a}$ = 0.6</td>
</tr>
<tr>
<td>$R_3 = \frac{\sum_{k=1}^{P} (f_a + f_{a3}) - f_3}{\sum_{k=1}^{P} f_a - \sum_{k=1}^{P} f_{a3}}$ = 0.3095</td>
</tr>
<tr>
<td>$MCEN(1) = 0$</td>
</tr>
<tr>
<td>$MCEN(3) = 0.2781$</td>
</tr>
<tr>
<td>$MCEN = \sum_{j=1}^{4} R_j$</td>
</tr>
</tbody>
</table>

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
DMCEN$_{id}$ 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


