Threshold optimisation for multi-label classifiers

Ignazio Pillai*, Giorgio Fumera, Fabio Roli

Department of Electrical and Electronic Engineering, University of Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
{*pillai, fumera, roli*}*@diee.unica.it*

Abstract

Many multi-label classifiers provide a real-valued score for each class. A well known design approach consists of tuning the corresponding decision thresholds by optimising the performance measure of interest. We address two open issues related to the optimisation of the widely used F measure and precision-recall (P-R) curve, with respect to the class-related decision thresholds, on a given data set. (i) We derive properties of the micro-averaged F, which allow its global maximum to be found by an optimisation strategy with a low computational cost. So far, only a suboptimal threshold selection rule and a greedy algorithm with no optimality guarantee were known. (ii) We rigorously define the macro- and micro-P-R curves, analyse a previously suggested strategy for computing them, based on maximising F, and develop two possible implementations, which can be also exploited for optimising related performance measures. We evaluate our algorithms on five data sets related to three different application domains.

Keywords: Multi-label classification, S-Cut thresholding, F measure, Precision-recall curve

1. Introduction

Multi-label (ML) classification problems occur in several applications like text categorisation, image annotation, and protein function prediction [1, 2, 3]. Contrary to the single-label problems, each sample can belong to more than one class. We denote the number of classes with N, and a sample with $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in X$ is a feature vector in a given feature space $X \subseteq \mathbb{R}^n$, and $\mathbf{y} \in Y = \{+1, -1\}^N$ encodes its class labels, where $y_k = +1(-1)$ means that $\mathbf{x}$ belongs (does not belong) to the k-th class. A ML classifier implements a decision function $f = (f_1, \ldots, f_N) : X \to Y$. In this work we consider the common case of classifiers that output a real-valued score for each class, $s_k(\mathbf{x}) \in \mathbb{R}$, and thus require a thresholding strategy to implement a decision function [1, 4, 3, 5]. Among thresholding strategies proposed so far, we focus on the widely used score-Cut (S-Cut) [4, 5]:

$$f_k(\mathbf{x}) = +1(-1), \quad \text{if} \quad s_k(\mathbf{x}) \geq t_k (< t_k), k = 1, \ldots, N \,, \tag{1}$$

where $t_1, \ldots, t_N$ are predefined threshold values, which should be computed by optimising the performance measure of interest on validation data. However, such optimisation turns out to be

*Corresponding author. Address: Department of Electrical and Electronic Engineering, University of Cagliari. Piazza d'Armi, 09123 Cagliari (Italy). Phone: +39 070 675 5817. Fax: +39 070 675 5782. E-mail: pillai@diee.unica.it (I.Pillai)

very hard for three widely used performance measures: micro- and macro-averaged precision and recall (P-R) curves, and micro-averaged Van Rijsbergen's F measure [6]. So far only a suboptimal threshold selection rule [4], and a greedy algorithm with no optimality guarantee [5], have been proposed for optimising the micro F. No author addressed the problem of computing the optimal micro- and macro-P-R curves as functions of S-Cut thresholds instead, except for a possible strategy which was only suggested in [4], and no rigorous definition even exists for such curves. Consequently, choosing an operational point according to a desired trade-off between P and R, as well as optimising related measures like average precision and break-even point, is still an open issue.

After a description of the considered performance measures and of related works (Sect. 2), we give the following contributions. First, in Sect. 3 we prove two properties of micro F as a function of S-Cut thresholds and exploit them to derive an optimisation strategy that guarantees to provide the global maximum with an upper bound of $O(n^2N^2)$ on computational complexity. We develop the corresponding optimisation algorithm, and discuss its relationship with the one of [5]. These results have been reported in our previous work [7]. Second, in Sect. 4 we rigorously define the macro- and micro-P-R curves, and analyse the strategy suggested in [4] for computing them. We develop two implementations of this strategy: one for computing the whole P-R curves and another one for obtaining a single point of such curves according to a desired trade-off between precision and recall. We also show how such strategy can be exploited to optimise related measures.

In Sect. 5 the proposed algorithms are experimentally evaluated on five benchmark data sets related to three different application domains. Conclusions and directions for future works are finally given in Sect. 6.

2. Background and previous works

In information retrieval, *precision* is the probability that a retrieved sample is relevant to a given query, and *recall* is the probability to retrieve a relevant sample. In multi-label classification each class corresponds to a distinct query. Precision and recall for the k-th class are then given respectively by $p_k = \mathrm{P}(y_k = +1 \mid f_k(\mathbf{x}) = +1)$, and $r_k = \mathrm{P}(f_k(\mathbf{x}) = +1 \mid y_k = +1)$. Empirical estimates of p_k and r_k can be obtained from a data set as: $\hat{p}_k = TP_k/(TP_k + FP_k)$, and $\hat{r}_k = TP_k/(TP_k + FN_k)$, where TP_k, FP_k and FN_k denote respectively the number of true positive, false positive and false negative samples. The widely F measure combines p_k and r_k into a scalar [6]:

$$F_{\beta,k} = (1 + \beta^2)/(1/p_k + \beta^2/r_k) \,, \tag{2}$$

while the parameter $\beta \in [0, +\infty]$ defines a trade-off between p_k and r_k.

Precision, recall and F over all classes are defined in two different ways, as empirical averages [1, 2]. *Macro-averaging* (denoted with the superscript 'M') consists of averaging the corresponding class-related measure:

$$\hat{p}^{\mathrm{M}} = \frac{1}{N}\sum_{k=1}^{N} \hat{p}_k, \quad \hat{r}^{\mathrm{M}} = \frac{1}{N}\sum_{k=1}^{N} \hat{r}_k \,, \tag{3}$$

$$\hat{F}^{\mathrm{M}}_{\beta} = \frac{1}{N}\sum_{k=1}^{N} \hat{F}_{\beta,k} = \frac{1}{N}\sum_{k=1}^{N} (1 + \beta^2)/\left((1 + \beta^2) + \frac{FP_k + \beta^2 FN_k}{TP_k}\right) . \tag{4}$$

It equally weights each class, and thus tends to be dominated by the performance on rare classes, which is usually lower than in common ones [4]. *Micro-averaging* (denoted with 'm') consists of computing precision and recall with respect to the sum of TP, FP and FN values over all classes:

$$\hat{p}^{\mathrm{m}} = \frac{\sum_{k=1}^{N} TP_k}{\sum_{k=1}^{N} (TP_k + FP_k)}, \quad \hat{r}^{\mathrm{m}} = \frac{\sum_{k=1}^{N} TP_k}{\sum_{k=1}^{N} (TP_k + FN_k)}, \tag{5}$$

$$\hat{F}_{\beta}^{\mathrm{m}} = \frac{1+\beta^2}{1/\hat{p}^{\mathrm{m}} + \beta^2/\hat{r}^{\mathrm{m}}} = (1+\beta^2)/\left((1+\beta^2) + \frac{\sum_{k=1}^{N}(FP_k + \beta^2 FN_k)}{\sum_{k=1}^{N} TP_k}\right). \tag{6}$$

The choice between the two averaging strategies is application-specific.

Consider now the problem of finding the values of S-Cut thresholds that maximise $\hat{F}_{\beta}^{\mathrm{m}}$ on a data set of n samples $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{n}$, given the corresponding scores $s_k(\mathbf{x}_i)$, $k = 1, \ldots, N$, of a trained ML classifier. We denote with $s_k(\mathbf{x}_{(i)})$ the scores sorted in increasing order, such that $s_k(\mathbf{x}_{(i)}) \leq s_k(\mathbf{x}_{(i+1)})$, $i = 1, \ldots, n-1$. We also denote $-\infty$ and $+\infty$ respectively as $s_k(\mathbf{x}_0)$ and $s_k(\mathbf{x}_{(n+1)})$. We finally denote with $\mathbf{t} = (t_1, \ldots, t_N)$ a specific value of the S-Cut thresholds, and with $\mathbf{T} = (T_1, \ldots, T_N)$ the thresholds considered as variables. It is now easy to see that, using decision function (1), $\hat{F}_{\beta}^{\mathrm{m}}(\mathbf{T})$ is a piece-wise constant function: it is constant for any $t_k \in [s_k(\mathbf{x}_{(i)}), s_k(\mathbf{x}_{(i+1)}))$, $i = 0, \ldots, n$, being equal the other threshold values, and can exhibit discontinuities for $t_k = s_k(\mathbf{x}_i)$, $i = 1, \ldots, n$, $k = 1, \ldots, N$. Note that, a reasonable choice is $t_k = (s_k(\mathbf{x}_{(i)}) + s_k(\mathbf{x}_{(i+1)}))/2$ for $1 \leq i < n$, while the choice is not straightforward for $[s_k(\mathbf{x}_{(0)}), s_k(\mathbf{x}_{(1)}))$ and $[s_k(\mathbf{x}_{(n)}), s_k(\mathbf{x}_{(n+1)}))$. In this work we shall not distinguish between $\mathbf{t}$ values that provide the same $\hat{F}_{\beta}^{\mathrm{m}}$ value.

The optimisation problem of interest can now be formalised as

$$\mathbf{t} = \arg\max \hat{F}_{\beta}^{\mathrm{m}}(\mathbf{T}). \tag{7}$$

However, since $\hat{F}_{\beta}^{\mathrm{m}}$ can not be decomposed over classes, i.e., into independent functions of the individual T_k's, solving (7) is non-trivial. So far, this issue has been addressed only in [4, 5]. In [4] the same threshold values that maximise $\hat{F}_{\beta}^{\mathrm{M}}$ were simply used. Although this choice is clearly suboptimal, its computational cost is negligible, since $\hat{F}_{\beta}^{\mathrm{M}}$ is additive over classes. In [5] a greedy optimisation algorithm was proposed. It consists of iteratively scanning $T_1, \ldots, T_N$, and updating each T_k to $\arg\max_{T_k} \hat{F}_{\beta}^{\mathrm{m}}(t_1, \ldots, t_{k-1}, T_k, t_{k+1}, \ldots, t_N)$, until the increase attained in the last scan is below a given value. In [5], this optimisation strategy was empirically shown to be more effective than using the thresholds that maximise $\hat{F}_{\beta}^{\mathrm{M}}$, but no optimality guarantee was provided.

In [4] a technique was also proposed against overfitting, named FBR. It was argued that overfitting is more likely for rarer classes, and that too low threshold values should hurt both $\hat{F}_{\beta}^{\mathrm{m}}$ and $\hat{F}_{\beta}^{\mathrm{M}}$, while too high values should hurt mainly $\hat{F}_{\beta}^{\mathrm{M}}$, and by a lower extent. Accordingly, it was proposed to set the threshold of a rare class either to $T_k = +\infty$ (FBR.0), or to the score of the top-ranked sample, $T_k = s_k(\mathbf{x}_{(n)})$ (FBR.1). Rare classes were defined as the ones for which $\hat{F}_{\beta,k} < fbr$, where fbr is a predefined value. In [5], FBR was found to be effective only for $\hat{F}_{\beta}^{\mathrm{M}}$.

Finally, to our knowledge no author addressed the problem of computing the macro- and micro-P-R curves of a ML classifier, as well as related measures like average precision and break-even point, when S-Cut is used. A formal definition of these curves has not even been given. In [4] it was suggested that the macro- and micro-P-R curves could be computed by maximising the corresponding F measure for different β values. However, this strategy was not further analysed, and no practical implementation was proposed either (e.g., which β values should be used).

3. Optimisation of the micro-averaged F measure

Here we state two properties related to the global maximum of $\hat{F}^{\mathrm{m}}_{\beta}(\mathbf{T})$, computed on a given data set. Their proof is reported in the Appendix. We then show that they allow one to solve problem (7) by an iterative optimisation strategy with an upper bound on computational complexity given by $O(n^2N^2)$. We also develop a possible implementation of such strategy.

Property 1. *Consider any given set of threshold values* $\mathbf{t} = (t_1, \ldots, t_N)$*. If, for each* $k = 1, \ldots, N$*,* $\max_{T_k} \hat{F}^{\mathrm{m}}_{\beta}(t_1, \ldots, t_{k-1}, T_k, t_{k+1}, \ldots, t_N) = \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t})$*, then* $\max_{\mathbf{T}} \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{T}) = \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t})$*.*

By contraposition, Property 1 implies that, if $\max_{\mathbf{T}} \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{T}) \neq \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t})$, then there exist at least one $k \in \{1, \ldots, N\}$ and one value $t'_k \neq t_k$, such that $\hat{F}^{\mathrm{m}}_{\beta}(t_1, \ldots, t_{k-1}, t'_k, t_{k+1}, \ldots, t_N) > \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t})$. This guarantees that $\max_{\mathbf{T}} \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{T})$ can be found by repeatedly updating each T_k to any value that provides an increase of $\hat{F}^{\mathrm{m}}_{\beta}$, while keeping fixed all the other T_j's, $j \neq k$, until no increase can be attained by changing any of the T_k's. It is now easy to see that the greedy algorithm of [5] (see Sect. 2) is a possible implementation of the above optimisation strategy, and thus actually provides the global maximum of $\hat{F}^{\mathrm{m}}_{\beta}$, provided that no early-stopping condition, as the one in[5], is used.

Let $t_1, \ldots, t_N$ be the threshold values at any step of the optimisation strategy sketched above. This strategy requires one to find, for any given k, a value $t'_k \neq t_k$ (if any) such that $\hat{F}^{\mathrm{m}}_{\beta}(t_1, \ldots, t_{k-1}, t'_k, t_{k+1}, \ldots, t_N) > \hat{F}^{\mathrm{m}}_{\beta}(t_1, \ldots, t_N)$. This problem can be solved by exhaustive search over the $n + 1$ possible values of T_k (this is also the strategy used in [5]). The following property shows that a lower number of values of T_k can be considered.

Property 2. *Consider any set of threshold values* $\mathbf{t} = (t_1, \ldots, t_N)$*, such that, for a given* k*,* $t_k = \arg\max_{T_k} F^{\mathrm{m}}_{\beta}(t_1, \ldots, t_{k-1}, T_k, t_{k+1}, \ldots, t_N)$*. If there exists another set* $\mathbf{t}' = (t'_1, \ldots, t'_{k-1}, t_k, t'_{k+1}, \ldots, t'_N)$*, such that* $F^{\mathrm{m}}_{\beta}(\mathbf{t}') > F^{\mathrm{m}}_{\beta}(\mathbf{t})$*, then the following is always true:* $F^{\mathrm{m}}_{\beta}(\mathbf{t}') = \max_{T_k \leq t_k} F^{\mathrm{m}}_{\beta}(t'_1, \ldots, t'_{k-1}, T_k, t'_{k+1}, \ldots, t'_N)$*.*

This means that, if our optimisation strategy is implemented by updating each threshold to a value which *locally maximises* F^{m}_{β}, then such value is guaranteed to be *not lower* than the current one. Clearly, this requires to initially set each T_k in $[s_k(\mathbf{x}_{(0)}), s_k(\mathbf{x}_{(n+1)}))$.

A possible implementation of our optimisation strategy, based on Properties 1 and 2, is given as Algorithm 1. In the Appendix D we prove that an upper bound on the number of `for` loops is given by $\frac{1}{2}\left(N^2n^2 + Nn\right) = O(n^2N^2)$. The experiments of Sect. 5 show that the actual cost can be much lower.

4. The macro- and micro-averaged precision-recall curves

In this section we deal with the problem of defining and computing the macro- and micro-P-R curves of a trained classifier, as functions of $T_1, \ldots, T_N$. To simplify the notation, we will write $\hat{p}$, $\hat{r}$ and $\hat{F}$, with no superscripts, to denote both the micro- and macro-averaged measures.

The P-R curve of a single class can be simply obtained by scanning the scores $s_k(\mathbf{x}_i)$ sorted in increasing or decreasing order: one obtains a set of up to $n + 1$ distinct pairs of values $(\hat{p}_k(T_k), \hat{r}_k(T_k))$, including $(0, 1)$ and $(1, 0)$ (see Fig. 1, left). Note that the dominated points (if any) can be discarded, as they provide both P and R values lower or equal to some other point.

Algorithm 1 $\hat{F}^{\mathrm{m}}_{\beta}$ maximisation algorithm, based on Properties 1 and 2.

Input: a set of scores values $s_k(\mathbf{x}_i)$, $i = 1, \ldots, n$, $k = 1, \ldots, N$
Output: a set of threshold values $t_1, \ldots, t_N$

$t_k \leftarrow$ any value in $(-\infty, s_k(\mathbf{x}_{(1)}))$, $k = 1, \ldots, N$
repeat
  $updated \leftarrow$ False
  for $k = 1, \ldots, N$ **do**
    $t_k^* \leftarrow \arg\max_{T_k \geq t_k} \hat{F}^{\mathrm{m}}_{\beta}(t_1, \ldots, t_{k-1}, T_k, t_{k+1}, \ldots, t_N)$
    if $\hat{F}^{\mathrm{m}}_{\beta}(t_1, \ldots, t_{k-1}, t_k^*, t_{k+1}, \ldots, t_N) > \hat{F}^{\mathrm{m}}_{\beta}(t_1, \ldots, t_N)$
    then $t_k \leftarrow t_k^*$, $updated \leftarrow$ True **end if**
  end for
until $updated$ = False
return $t_1, \ldots, t_N$

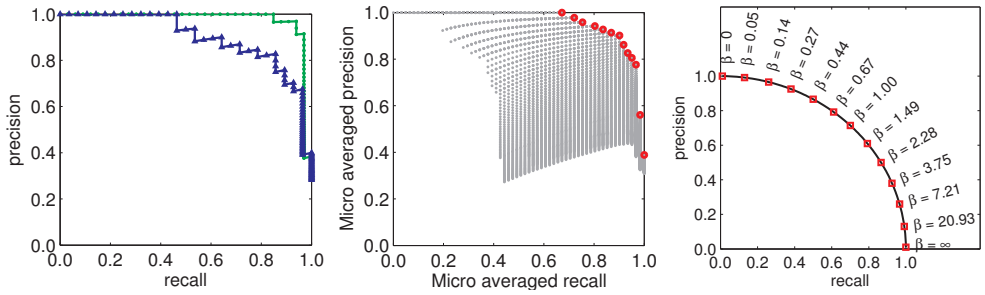

Figure 1: Left: P-R curve for two different classes, evaluated on $n = 100$ samples of the Reuters data set (see Sect. 5). Middle: micro P-R points of the above classes (Eq. (5)), corresponding to the 101×101 distinct pairs of threshold values. Highlighted points correspond to the micro-P-R curve of Definition 1. Right: hypothetical P-R curve corresponding to the arc of circumference of radius 1 and centred in the origin; the thirteen points that maximise F_β are shown, for $\beta = (\cot \alpha)^{3/2}$, $\alpha = \frac{k}{12}\frac{\pi}{2}$, $k = 0, \ldots, 12$.

Often, a piece-wise constant approximation of the P-R curve is obtained by computing the *interpolated precision*: for each $r_k \in [0, 1)$, it is defined as $\max_{\hat{r}_k \geq r_k} \hat{p}_k$, where each $(\hat{p}_k, \hat{r}_k)$ belongs to the discrete P-R curve [8].

Defining the macro- and micro-P-R curves is not straightforward, instead. Indeed, computing $\hat{p}(\mathbf{T})$ and $\hat{r}(\mathbf{T})$ for all possible $\mathbf{T}$ values, one obtains up to $(n+1)^N$ distinct P-R points which do not belong to a curve (see Fig. 1, middle). However, after discarding dominated points one obtains a sequence of strictly decreasing recall and strictly increasing precision, including $(0, 1)$ and $(1, 0)$, as in the case of a single class (see again Fig. 1, middle). Accordingly, we propose the following, formal definition:

Definition 1. *When S-Cut is used, the macro- and micro-P-R curves are defined, on a given set of n samples, as the subset of non-dominated points in* $\{(\hat{p}(\mathbf{t}), \hat{r}(\mathbf{t}))\}_{\mathbf{t} \in \mathbb{R}^N}$.

4.1. Computing the macro- and micro-averaged precision-recall curves

Computing the micro- and macro-P-R curves by exhaustive search, by naively applying Definition 1, is clearly impractical. The strategy suggested in [4], based on maximising the corresponding F measure for different β values, is a potentially better alternative. Indeed, it was

Algorithm 2 Approximated macro/micro-P-R curve.

Input: A set of scores values $s_k(\mathbf{x}_i)$, $k = 1, \ldots, N$, $i = 1, \ldots, n$; the desired number $2M + 1$ of P-R points
Output: A set C of up to $2M + 1$ P-R points

$C \leftarrow \emptyset,\ r_{\max} \leftarrow 0,\ p_{\text{prev}} \leftarrow 1,\ r_{\text{prev}} \leftarrow 0$
for $k = 1, \ldots, 2M$ **do**
  if $k = 2M$ **then** $\hat{p} \leftarrow 0,\ \hat{r} \leftarrow 1$
  else $\beta \leftarrow \left(\cot \frac{k}{2M}\frac{\pi}{2}\right)^{3/2}$, $\mathbf{t} \leftarrow \arg\max_{\mathbf{T}} \hat{F}_\beta$ **end if**
  if $\hat{p}(\mathbf{t}) \neq p_{\text{prev}}$ **and** $\hat{r}(\mathbf{t}) > r_{\max}$ **then** $C \leftarrow C \bigcup \{(p_{\text{prev}}, r_{\text{prev}})\},\ r_{\max} \leftarrow r_{\text{prev}}$ **end if**
  $r_{\text{prev}} \leftarrow \hat{r}(\mathbf{t}),\ p_{\text{prev}} \leftarrow \hat{p}(\mathbf{t})$
end for
return C

known that maximising macro F is computationally cheap, and our results in Sect. 3 have shown that the same holds for micro F. Accordingly, in the following we analyse such strategy. All proofs are reported in the Appendix.

We first investigate whether the above strategy provides *all* the P-R points of Definition 1 (completeness), and *only* such points (optimality).

Property 3 (completeness). *For any given* $\beta \in [0, +\infty)$*, if* $\mathbf{t} = \arg\max_{\mathbf{T}} \hat{F}^{\text{m}}_\beta(\mathbf{T})$*, then* $(\hat{p}^{\text{m}}(\mathbf{t}), \hat{r}^{\text{m}}(\mathbf{t}))$ *belongs to the micro-P-R curve. Instead, if* $\mathbf{t} = \arg\max_{\mathbf{T}} \hat{F}^{\text{M}}_\beta(\mathbf{T})$*, then* $\left(\hat{p}^{\text{M}}(\mathbf{t}), \hat{r}^{\text{M}}(\mathbf{t})\right)$ *may not belong to the macro-P-R curve.*

Property 4 (optimality). *There can be some point* $(\hat{p}, \hat{r})$ *belonging to the macro or micro-P-R curve, such that* $(\hat{p}, \hat{r}) \neq (\hat{p}(\mathbf{t}), \hat{r}(\mathbf{t}))$*, for all* $\mathbf{t}$ *that can be obtained as* $\mathbf{t} = \arg\max_{\mathbf{T}} \hat{F}_\beta(\mathbf{T})$.

This means that the strategy of [4] is not complete, and is optimal only for the micro-averaged P-R curve.

Consider now the practical issue, not addressed in [4], of choosing a finite set of β values to obtain the (approximated) P-R curve using the above strategy. Note first that the points $(0, 1)$ and $(1, 0)$, and the break-even point $p = r = \frac{1}{\sqrt{2}}$, correspond respectively to $\beta = 0, \beta = +\infty$, and $\beta = 1$. Ideally, the chosen β values should lead to P-R points that cover as much as possible the whole range of P-R values, and are distributed as uniformly as possible along the curve. To this aim, let us consider the hypothetical P-R curve corresponding to the arc of circumference of radius 1, centred in the origin. Its points can be expressed as $(p, r) = (\sin\alpha, \cos\alpha)$, where α is the angle spanned with respect to the axis $r = 0$. It is easy to see that each of these points can be obtained maximising the F_β expression (2) for $\beta = (\cot\alpha)^{3/2}$ (by replacing p and r with $\sin\alpha$ and $\cos\alpha$ in (2), and setting to zero the derivative with respect to β). Accordingly, one can obtain $2M + 1$ points (for any non-negative integer M) uniformly distributed along the above curve, including $(0, 1)$, $(1, 0)$ and the break-even point $(\frac{1}{\sqrt{2}}, \frac{1}{\sqrt{2}})$, by choosing $\beta = \cot\frac{k}{2M}\frac{\pi}{2}$, $k = 0, 1, \ldots, 2M$. Note that these β values are non-uniformly distributed in a linear scale (see Fig. 1).

In practice, the micro- and macro-P-R curve may be different from the above one, and it is not guaranteed that maximising $\hat{F}_\beta$ provides a point corresponding to $\alpha = \cot^{-1}(\beta^{2/3})$. Furthermore, different β values may lead to the same P-R point. Nevertheless, the above result suggests a very simple and well grounded choice of β values. Algorithm 2 formalises the corresponding procedure for approximating the macro- and micro-P-R curves. Its computational cost is $2M + 1$ times the cost of maximising $\hat{F}_\beta$.

Algorithm 3 Maximising precision under the constraint $\hat{r} \geq r_{\min}$.

Input: A set of scores values $s_k(\mathbf{x}_i)$, $i = 1, \ldots, n$, $k = 1, \ldots, N$; the minimum allowed recall, r_{min}; the maximum number of iterations, $I_{\max}$; the minimum variation of precision, $\Delta p_{\min}$

Output: A set of threshold values $t_1, \ldots, t_N$

$i \leftarrow 0,\ \alpha_{\min} \leftarrow 0,\ \alpha_{\max} \leftarrow \pi/2,\ p_{\text{best}} \leftarrow 0,\ \Delta p \leftarrow +\infty$
while $i \leq I_{\max}$ **and** $\Delta p > \Delta p_{\min}$ **do**
  $i \leftarrow i + 1,\ \alpha \leftarrow (\alpha_{\min} + \alpha_{\max})/2,\ \beta \leftarrow (\cot \alpha)^{3/2}, \mathbf{t} \leftarrow \arg\max_{\mathbf{T}} \hat{F}_\beta(\mathbf{T})$
  if $\hat{r}(\mathbf{t}) < r_{\min}$ **then** $\alpha_{\max} \leftarrow \alpha$ **else**
    $\alpha_{\min} \leftarrow \alpha,\ \Delta p \leftarrow |\hat{p}(\mathbf{t}) - p_{\text{best}}|$
    if $\hat{p}(\mathbf{t}) > p_{\text{best}}$ **then** $p_{\text{best}} \leftarrow \hat{p}(\mathbf{t}), \mathbf{t}^* \leftarrow \mathbf{t}$ **end if**
  end if
end while
return $\mathbf{t}^*$

4.2. Selecting an operational point on the precision-recall curve

Computing the P-R curve can be an intermediate step towards selecting one or a subset of its points, with two possible goals. One is to choose an operational point according to the application requirements expressed as a trade-off between P and R, e.g., maximising P (R) under the constraint that R (P) is higher or equal to a given value, and finding both P and R higher or equal to given values. Another one is to compute a different performance measure related to the P-R curve, like the average precision, or the break-even point. To this aim, one could first compute the P-R curve using Algorithm 2, and then select the desired point(s). However, this may require one to compute beforehand an unnecessarily high number of P-R points.

We propose here an alternative strategy analogous to binary search, which was inspired by the one proposed in [9] for setting the operational point on the ROC curve. It consists of iteratively searching for the β value(s) that provides the desired point(s), with a given degree of approximation, without constructing the whole P-R curve beforehand. Such strategy exploits the criterion to select β values proposed in Sect. 4.1, and the following property, which guarantees that maximising $\hat{F}_\beta$ for increasing β values provides P-R points with non-decreasing recall and non-increasing precision.

Property 5. *Consider any* $\beta_1 \geq \beta_2$*, and let* $\mathbf{t}^{(i)} = \arg\max_{\mathbf{T}} \hat{F}_{\beta_i}(\mathbf{T})$*,* $i = 1, 2$*. The following always holds for the corresponding P-R points:*
$\hat{p}(\mathbf{t}^{(1)}) \leq \hat{p}(\mathbf{t}^{(2)})$*,* $\hat{r}(\mathbf{t}^{(1)}) \geq \hat{r}(\mathbf{t}^{(2)})$*.*

We shall only consider the following application requirement, since all the other cases can be dealt with similarly:

$$\max_{\mathbf{T}} \hat{p}, \quad \text{s.t. } \hat{r} \geq r_{\min} \,. \tag{8}$$

Property 5 implies that a (unknown) β^* value exists, such that any $\beta > \beta^*$ leads to a P-R point with $\hat{r} < r_{\min}$, which violates the constraint in (8), while any $\beta < \beta^*$ leads to $\hat{r} \geq r_{\min}$ and $\hat{p} \leq \hat{p}^*$, where $\hat{p}^*$ is the value provided by β^*. The solution of problem (8) is thus obtained by maximising $\hat{F}_{\beta^*}$. The value of β^* can be approximated by a kind of binary search strategy. An interval $[\alpha_{\min}, \alpha_{\max}]$ is considered, initially set to $[0, \pi/2]$. At each iteration, the midpoint of the current interval is evaluated, $\alpha = (\alpha_{\min} + \alpha_{\max})/2$, and the point $(\hat{p}(\mathbf{t}), \hat{r}(\mathbf{t}))$, corresponding to $\arg\max_{\mathbf{T}} \hat{F}_\beta$, is computed, where $\beta = (\cot \alpha)^{3/2}$ (see Sect. 4.1). If $\hat{r}(\mathbf{t}) < r_{\min}$ ($\hat{r}(\mathbf{t}) \geq r_{\min}$), then $\alpha_{\max}$ ($\alpha_{\min}$) is updated to α. A possible implementation is given as Algorithm 3: the search is

Dataset	Samples (train/test)	Features	Classes	Class frequency (min/max)	Labels per sample (mean/max)
Reuters	7769 / 3019	18157	90	1E-4/0.37	1.23/15
Ohsumed	12775 / 3750	17341	99	2E-4/0.25	1.49/11
RCV1v2	3000 / 3000	47237	101	3E-4/0.46	3.19/12.4
Yeast	1500 / 917	104	14	0.06/0.75	4.23/11
Scene	1211 / 1196	295	6	0.14/0.23	1.06/3

Table 1: Characteristics of the five data sets used in the experiments, estimated on the training set. For RCV1v2, average values over the five training sets are reported.

stopped after a predefined number of $I_{\max}$ iterations, or when two precision values closer than a predefined threshold $\Delta p_{\min}$ are obtained in two successive steps, with a recall not lower than $r_{\min}$.

5. Experiments

We experimentally evaluated Algorithms 1–3 on five benchmark ML data sets related to three different domains: Reuters 21578, Reuters RCV1v2 [10], and the Heart Disease sub-tree of Ohsumed (text categorisation) [11, 12]; Scene (image annotation) [3]; and Yeast (gene annotation) [13].

All data sets are originally subdivided into a training and a testing set, except for RCV1v2, for which five different pairs of training and testing sets are available. For RCV1v2, Yeast and Scene we used the feature vectors of [5]. For data sets related to text categorisation we used the *bag-of-words* feature model; we first carried out stemming, stop-word removal, and a further feature selection step using the information gain criterion [1]. The main characteristics of the data sets are reported in Table 1.

We implemented ML classifiers by independently training a binary classifier for each class (*binary relevance*) [1, 2]. We considered three statistical classifiers widely used in ML tasks: support vector machines (SVM) [14] with RBF kernel for Scene and Yeast, and lineare kernel for the other data sets; k-nearest neighbours (k-NN) [15]; and Naive Bayes (NB). The latter was not used for Scene, due to its poor performance. For text categorisation data sets we used tf–idf features for SVM and k-NN, and Boolean features with the multi-variate Bernoulli model of [16] for NB. NB was implemented with equal-width feature discretisation [17] for Scene. Selection of features and of classifier parameters was carried out through a four-fold cross-validation (C-V) on the original training set (the first training set was used for RCV1v2).

5.1. Optimisation of the micro-averaged F measure

We evaluated the computational cost and the tendency to overfit of Algorithm 1, extending the experiments reported in [7]. The latter is an obvious concern, as Algorithm 1 maximises F^{m} on a given data set without any countermeasure against overfitting. We did not make any comparative evaluation of its performance, since the only alternatives are the ones of [4, 5]. The former was shown to be less effective in [5], while we proved that the latter is equivalent to Algorithm 1, provided that no early-stopping condition is used.

In our experiments we considered only $\hat{F}^{\mathrm{m}}_1$ ($\beta = 1$), as in [4, 5]. Algorithm 1 was applied to the union of the testing scores obtained by a five-fold C-V, which was carried out on the training samples of each run of the experiments. Five runs of the experiments were carried out for RCV1v2, on the five pairs of training and testing sets. Ten runs were carried out on the other

	Reuters	Ohsumed	RCV1v2	Yeast	Scene
$C(N,n)/[N(n+1)]$	1.13 ± 0.0	1.46 ± 0.04	1.25 ± 0.03	1.94 ± 0.17	1.46 ± 0.05
Iter. over all classes	4.0 ± 0.0	4.6 ± 0.5	4.2 ± 0.4	3.3 ± 0.5	3.1 ± 0.3

Table 2: Average computational cost of Algorithm 1 over the different runs of the experiments, using the SVM classifier. $C(N,n)$ denotes the number of `for` loops.

data sets: their original training set was split into ten disjoint subsets, and eight of them were used as the training set in each run.

5.1.1. Computational cost

We denote the number of `for` loops carried out by Algorithm 1 as $C(N,n)$, to highlight its dependence on the number of classes (N), and of samples (n). To evaluate computational cost in a comparable way across data sets, that exhibit different values of N and n, in the first row of Table 2 we report the average ratio (over the different runs of the experiments) of $C(N,n)$ to the maximum number of `for` loops in a *single* `repeat-until` iteration, which is given by $N(n+1)$. Reported values refer to the SVM classifier. Similar results were observed with the k-NN and NB classifiers. Table 2 shows that the actual computational cost was always less than twice the maximum cost of a single `repeat-until` iteration, namely $C(N,n) < 2N(n+1)$, which is much lower than the upper bound of $\frac{1}{2}\left(N^2n^2 + Nn\right)$. This provides evidence that Algorithm 1 can scale very well on large data sets with many classes.

The second row of Table 2 shows that the number of `repeat-until` iterations was between 3 and 5. While the algorithm of [5] carries out exactly $N(n+1)$ `for` iterations at each `repeat-until` iteration, a reduction of 30% to 70% was attained by Algorithm 1, thanks to Property 2.

Furthermore, a single `repeat-until` iteration was suggested for the algorithm of [5], independently on the order in which the classes are scanned in the `for` loop. We further investigated this issue, by evaluating $\hat{F}_1^{\mathrm{m}}$ on the same data used by Algorithm 1, after each `for` loop. In Fig. 2 we report that the results for the Scene and Ohsumed data sets, attained by scanning the classes in three different orders: the same order in which they are listed in the data set, and for increasing ("Incr.") and decreasing ("Decr.") class frequency. For Scene, one iteration was sufficient to approach the global maximum of $\hat{F}_1^{\mathrm{m}}$, except for the "no ordering" case. However, two iterations were necessary for Ohsumed, while the value of $\hat{F}_1^{\mathrm{m}}$ after the first iteration was much lower than the global maximum. Furthermore, such value strongly depends on the order in which thresholds are scanned. This clearly shows that more than one *repeat-until* loop may be required.

5.1.2. Overfitting

To evaluate whether and to what extent overfitting occurs, we compared the value of $\hat{F}_1^{\mathrm{m}}$ on the same data used by Algorithm 1, and the corresponding value on testing data. These results, reported respectively in the third and fourth columns of Table 3, show that lower values were attained on testing data. On the other hand, running Algorithm 1 on testing data (only for the purposes of this experiment), we observed that the values of $\hat{F}_1^{\mathrm{m}}$ are very close to the ones on validation data. This confirms that some overfitting occurred. Its amount, i.e., the difference between the $\hat{F}_1^{\mathrm{m}}$ values in the third and fourth column of Table 3, was nevertheless rather limited: it was almost always below 0.03, except for RCV1v2. The highest amount of overfitting occurred in text categorisation data sets, that exhibit the largest number of classes and a strong class imbalance (see Table 1).

Data set	Classifier	Validation	Test	Test+FBR.0	Test+FBR.1
Reuters	SVM	0.907±0.001	0.880±0.002	0.879±0.002	0.878±0.002
	k-NN	0.854±0.002	0.825±0.003	0.825±0.003	0.824±0.003
	NB	0.767±0.003	0.747±0.003	0.747±0.003	0.747±0.003
Ohsumed	SVM	0.703±0.002	0.683±0.002	0.683±0.002	0.683±0.002
	k-NN	0.600±0.002	0.574±0.003	0.574±0.003	0.574±0.003
	NB	0.529±0.001	0.502±0.002	0.502±0.002	0.502±0.002
RCV1v2	SVM	0.811±0.005	0.775±0.003	0.775±0.003	0.775±0.003
	k-NN	0.785±0.005	0.746±0.003	0.745±0.003	0.745±0.003
	NB	0.711±0.003	0.657±0.003	0.657±0.003	0.628±0.055
Yeast	SVM	0.682±0.002	0.678±0.003	0.678±0.003	0.678±0.003
	k-NN	0.667±0.003	0.661±0.003	0.661±0.003	0.660±0.002
	NB	0.625±0.005	0.619±0.003	0.619±0.003	0.579±0.018
Scene	SVM	0.778±0.007	0.769±0.006	0.769±0.006	0.769±0.006
	k-NN	0.739±0.007	0.711±0.004	0.711±0.004	0.711±0.004

Table 3: Average value and standard deviation of $\hat{F}_1^{\mathrm{m}}$ over the different runs of the experiments, attained on validation data ("Validation" column), and on testing data (the three right-most columns), without using FBR, using FBR.0 and using FBR.1.

Data set (class frequency)	Validation	Test	Test+FBR.0	Test+FBR.1
Reuters - A (High)	0.937±0.001	0.928±0.002	0.928±0.002	0.928±0.002
Reuters - B (Low)	0.839±0.007	0.767±0.004	0.767±0.004	0.767±0.004
Reuters - C (Very low)	0.744±0.014	0.651±0.016	0.651±0.016	0.647±0.016

Table 4: Average value and standard deviation of $\hat{F}_1^{\mathrm{m}}$ over ten runs of the experiments, attained on the three subsets A, B and C of Reuters (see text) with the SVM classifier. See the caption of Table 3 for the meaning of columns.

We then evaluated whether the FBR heuristic is effective against overfitting. We estimated the value of the fbr parameter (see Sect. 2) through an inner five-fold C-V, which was carried out on each training fold of the outer C-V used for computing decision thresholds, similarl to [5]. The corresponding values of $\hat{F}_1^{\mathrm{m}}$ are reported in the two right-most columns of Table 3. In most cases using FBR did not affect the $\hat{F}_1^{\mathrm{m}}$ values. Even more, FBR.1 *worsened* the performance on the RCV1v2 and Yeast data sets, when the NB classifier was used.

We further investigated whether FBR can be effective, at least when all classes are very rare. We repeated the above experiments, using only the SVM classifier, on three data sets obtained from Reuters by considering three disjoint subsets of classes: (A): the 10 most populated classes, each of which contain at least 2% of the original samples; (B): the 22 classes that contain between 0.5% and 2% of the samples; (C): the remaining 58 rarer classes, that contain to less than 0.5% of the samples. In data sets A, B and C, no class label was assigned to samples not belonging to any of the considered classes (i.e., we set $\mathbf{y} = \{-1\}^N$ for such samples). The results are reported in Table 4. As expected, the difference between the $\hat{F}_1^{\mathrm{m}}$ values attained on validation and on testing data, without using FBR, is minimum for A, and maximum for C. However, also in this case the use of FBR did not provide any improvement.

The ineffectiveness of FBR observed in our experiments is in agreement with the results of [5]. A possible reason is that for maximising F_β^{m} it is crucial to reduce FP errors on rare classes (see Eq. 6), whose amount can be much higher than the one of FNs. This is attained by increasing the corresponding thresholds as much as possible, which is what FBR does, for rare classes. However, the values of such thresholds can be reliably estimated also from validation data (i.e., before FBR is applied), due to the relatively large number of negative samples, especially in rare

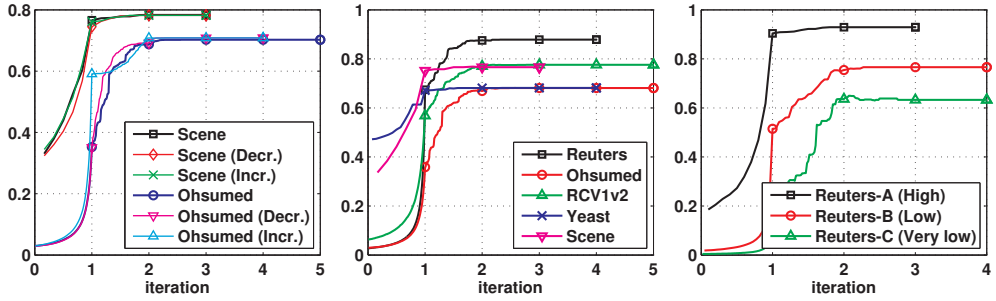

Figure 2: $\hat{F}_1^{\mathrm{m}}$ values as a function of the `repeat-until` loops made by Algorithm 1. Left: values attained on validation data on Scene and Ohsumed, by scanning the thresholds in three different orders (see text). Middle: values attained on the five original testing sets. Right: values attained on the A, B and C data sets obtained from Reuters (see text).

classes. Therefore, threshold values selected by FBR are likely to provide no improvement.

We finally investigated whether an early-stopping strategy, similar to the one used in the back-propagation learning algorithm for neural networks, can be more effective than FBR against overfitting. To this aim, we evaluated $\hat{F}_1^{\mathrm{m}}$ on testing data, after each `for` loop of Algorithm 1, to see if a maximum is reached before the algorithm stops. Note that, contrary to the back-propagation algorithm, Algorithm 1 is guaranteed to find no local maxima of $\hat{F}_1^{\mathrm{m}}$ (if any) on its input data. In Fig. 2 we report the results attained by the SVM classifier on the five original data sets (left plot), and on the A, B and C data sets obtained from Reuters (right plot). It can be seen that $\hat{F}_1^{\mathrm{m}}$ was almost always increasing. The only exception was data set C (which is made up of very rare classes), where the maximum $\hat{F}_1^{\mathrm{m}}$ was attained at the beginning of the third iteration. However, the subsequent decrease of $\hat{F}_1^{\mathrm{m}}$ is negligible. These results do not provide any evidence that overfitting can be prevented by early-stopping.

To sum up, Algorithm 1 seems to incur a non-negligible overfitting only in presence of a large number of very rare classes, and devising effective techniques against it remains an open issue.

5.2. Optimising the macro- and micro-averaged precision-recall curves

In this section we evaluate the strategy of [4] for computing the macro- and micro-P-R curves, to assess whether the choice of β values proposed in Sect. 4 provides uniformly distributed points along the curve, and whether and to what extent overfitting occurs. We then assess the computational cost of Algorithm 3. We used the same experimental setting of Sect. 5.1. The only difference is that only one run of the experiments was made using only the original training set (for RCV1v2 we used the first training/testing pair).

Fig. 3 shows the macro- and micro-P-R curves obtained on testing samples by maximising the corresponding $\hat{F}_\beta$ measure through Algorithm 1, using the SVM classifier. Similar results were obtained using the other classifiers. Twenty-one β values in the range $(0, +\infty)$ were considered, including $\beta = 1$. The resulting P-R points turned out to be distributed rather uniformly along the curve. A few exceptions can be observed, especially for Reuters and Yeast, in which recall did not approach zero. This is due to the fact that only dominated P-R points with lower recall values were obtained.

To evaluate overfitting, we compared the P-R curves attained on testing data, with the ones obtained on the input data of Algorithm 1. In Fig. 4 we report the results on Reuters, where

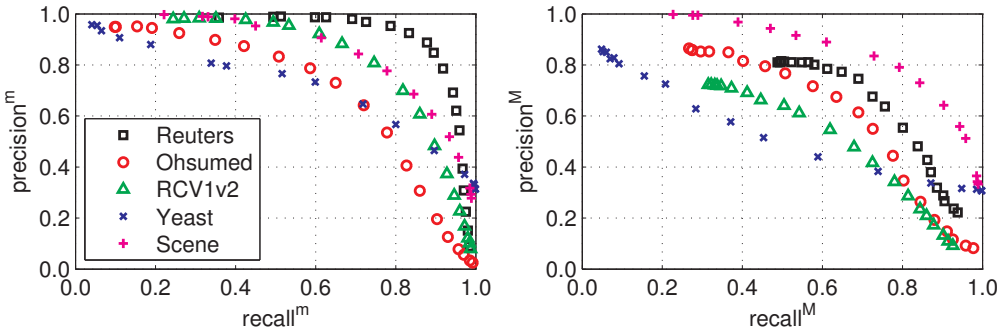

Figure 3: Micro and macro-P-R curves on the testing set, obtained by maximising the corresponding $\hat{F}_\beta$ measure for twenty-one different β values (see text).

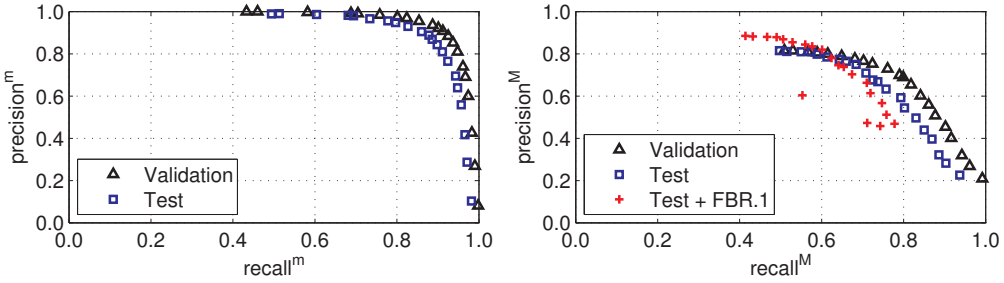

Figure 4: Micro and macro-P-R curves for Reuters, on testing and validation samples, for the same β values of Fig. 3. The macro-P-R curve obtained on testing data using FBR.1 is also shown ('+').

the highest difference between validation and testing data was observed. The difference between precision values, for the same β value, was not higher than 0.05 in the case of micro-averaging (Fig. 4, left), and than 0.07 for macro-averaging (Fig. 4, right). Similar differences were observed for recall. Overfitting was thus not negligible.

In Sect. 5.1 we did not find any effective technique for reducing overfitting, in the optimisation of micro F. Instead, in the case of macro-averaging FBR.1 was found effective in [5]. Accordingly, we evaluated it for the purpose of computing the macro-P-R curve. The results are shown in Fig. 4 (right), for $fbr = 0.1$. Similar results were obtained for different fbr values. The effect of FBR.1 was to move the P-R points toward higher precision and lower recall values. This was beneficial for the lowest recall values (i.e., $\beta < 1$), to the extent that some of the new P-R points dominated points obtained without using FBR.1. However, FBR.1 was highly detrimental when a high recall is required ($\beta > 1$): old P-R points with high recall were moved by FBR.1 to clearly suboptimal, dominated points, that sometimes were very far from the bottom-right region of the P-R plane. FBR seems thus effective for the macro-averaged P-R curve, only when a high precision is required.

Finally, we evaluated the computational cost of Algorithm 3, for solving problem (8) with a desired degree of approximation Δp_{min}. We set the maximum number of iterations to $I_{\max} = 100$, $\Delta p_{min} = 10^{-4}$, and run Algorithm 3 for 10 different r_{min} values: $0.10, 0.20, \ldots, 0.90, 0.99$. The results attained on the five data sets, using the SVM classifier, are reported in Table 5. Similar

	Reuters	Ohsumed	RCV1v2	Yeast	Scene
Micro-averaging	5/7.2/11	8/10.1/12	5/7.9/10	4/7.6/10	4/7.0/10
Macro-averaging	5/7.0/13	5/10.0/16	5/8.3/11	5/9.1/13	5/7.7/14

Table 5: Number of P-R points (minimum/average/maximum) explored by Algorithm 3, on the Reuters data set, over 10 different values of $r_{\min}$, using the SVM classifier.

results were obtained with the other classifiers. Algorithm 3 always found a solution with the desired degree of approximation, by evaluating up to 16 P-R points (i.e., β values) only. If the whole P-R curve had been constructed beforehand, a much higher number of β values should have been considered, to attain the same degree of approximation. This can be argued from Fig. 3, where 21 P-R points have been computed: the distance between the precision values of any pair of such points is always much higher than $\Delta p_{min} = 10^{-4}$.

6. Conclusions

In this paper we addressed two open issues of the S-Cut thresholding strategy, which is used for multi-label classifiers that output a real-valued score for each class: optimising the micro F measure, and computing the macro- and micro-averaged P-R curves, on a given data set, as a function of the S-Cut decision thresholds. We derived the properties of these performance measures, that allowed us: (i) to develop an optimisation algorithm which guarantees to find the global maximum of the micro F measure, on a given data set, with low computational cost; (ii) to analyse a strategy previously suggested to compute the macro- and micro-P-R curves, based on maximising the corresponding F measure for different β values, and to develop practical and computationally cheap implementations of such strategy.

Our results have several interesting follow-ups: (i) Maximising the micro F measure on a given data set can incur overfitting, especially for high unbalanced data sets. Suboptimal strategies proposed so far for selecting the S-Cut thresholds, as well as heuristic techniques to prevent overfitting, and an early stopping strategy assessed in this work, do not provide an effective solution. Developing effective strategies against overfitting remains thus an open problem. (ii) The accuracy of available multi-label classifiers, for a given task, may be not sufficiently high to meet application requirements, like the one considered in Sect. 4 for the P-R curve. Depending on the application at hand, we argue that a possible solution is using a *reject option*, namely, allowing the classifier to withhold assigning a class label, if it is deemed not sufficiently reliable. This is a well known technique to improve the accuracy of non-rejected decisions, for single-label problems. Its application to ML classifiers has however been considered only in preliminary works by the authors (see [18]), and still exhibits several challenging issues. (iii) The objective function of standard learning algorithms used in ML problems (e.g., SVMs), is not related to measures based on precision and recall. This is one of the reasons why such measures can be improved by tuning the decision thresholds of S-Cut. However, some works proposed modifications of standard objective functions, to approximate the F measure (e.g., [19]). It would thus be interesting to compare the performance of S-Cut with the one that can be attained using classifiers like the one proposed in [19]. (iv) Algorithm 2 provides a discrete approximation of P-R curves. If a continuous approximation is needed (e.g., for accurately estimating the break-even point), a proper interpolation procedure should be used. In [20], such a procedure was proposed for the P-R curve of a single class: it can be interesting to extend it to macro- and micro-P-R curves.

Acknowledgements

This work has been partly supported by grant from Regione Autonoma della Sardegna awarded to Ignazio Pillai, PO Sardegna FSE 2007-2013, L.R.7/2007 "Promotion of the scientific research and technological innovation in Sardinia" and by the project CRP-18293 funded by Regione Autonoma della Sardegna, L.R. 7/2007, Bando 2009.

Appendix A. Auxiliary equivalences.

The following equivalences are exploited in the next sections. Due to lack of space, their proof is reported in the online supplementary material.

Equivalence A.1. Given four real values A, B, ΔA and ΔB, with $B > 0$, $\Delta B < 0$, and $B + \Delta B > 0$, the following equivalence holds:

$$\frac{A + \Delta A}{B + \Delta B} < \frac{A}{B} \quad \Leftrightarrow \quad \frac{A}{B} < \frac{\Delta A}{\Delta B}. \tag{A.1}$$

Equivalences A.2 and A.3. Given four real values A, B, ΔA and ΔB, with $B > 0$, $\Delta B > 0$, the following equivalences hold:

$$\frac{\Delta A}{\Delta B} < \frac{A + \Delta A}{B + \Delta B} \quad \Leftrightarrow \quad \frac{A + \Delta A}{B + \Delta B} < \frac{A}{B}, \tag{A.2}$$

$$\frac{\Delta A}{\Delta B} < \frac{A}{B} \quad \Leftrightarrow \quad \frac{\Delta A}{\Delta B} < \frac{A + \Delta A}{B + \Delta B} < \frac{A}{B}. \tag{A.3}$$

Appendix B. Proof of Property 1

Consider a set of threshold values $\mathbf{t} = (t_1, \ldots, t_N)$, and another set obtained from $\mathbf{t}$ by changing the values of m thresholds, with $m \leq N$. Without losing generality, we assume that the first m thresholds are changed. We denote the latter set as $\mathbf{t}^{(1,\ldots,m)} = (t'_1, \ldots, t'_{m-1}, t'_m, t_{m+1}, \ldots, t_N)$. Let us also denote with $\mathbf{t}^{(k)}$ the threshold values obtained from $\mathbf{t}$ by changing only the k threshold from t_k to t'_k, $k \in \{1, \ldots, m\}$. In the following we prove that, for any given $m \in \{2, \ldots, N\}$ and $\mathbf{t}^{(1,\ldots,m)}$, the following implication holds:

$$\text{if} \quad \forall k \leq m \quad \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t}) > \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t}^{(k)}), \quad \text{then} \quad \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t}) > \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t}^{(1,\ldots,m)}) \tag{B.1}$$

Clearly, this implies that Property 1 is true.

Consider first Eq. (6). We denote the values $\sum_{k=1}^{N}(FP_k + \beta^2 FN_k)$ and $\sum_{k=1}^{N} TP_k$, corresponding to $\mathbf{t}$, respectively as E and TP. We also denote the same values, corresponding to $\mathbf{t}^{(k)}$, as $E + \Delta E_k$ and $TP + \Delta TP_k$. Since FP_k, FN_k and TP_k depend only on t_k, $\Delta E_k = \Delta TP_k = 0$, for any $k > m$.

From Eq. (6) it is easy to see that implication (B.1) can be rewritten as:

$$\text{if} \quad \forall k \leq m \quad \frac{E}{TP} < \frac{E + \Delta E_k}{TP + \Delta TP_k}, \text{ then} \quad \frac{E}{TP} < \frac{E + \sum_{k=1}^{m} \Delta E_k}{TP + \sum_{k=1}^{m} \Delta TP_k} \tag{B.2}$$

tIf $m = 1$, (B.2) is trivially true. If $m > 1$, we prove it by induction. First, we prove that it holds when $m = 2$. Then we prove that, if (B.2) holds for any $m = m^* \in \{2, \ldots, N-1\}$, then it holds also for $m = m^* + 1$.

Base case: $m = 2$. Assume that the consequent part of (B.2) is false, namely, a set of thresholds $\mathbf{t}^{(1,2)}$ exists, such that $F_\beta^{\mathrm{m}}(\mathbf{t}) < F_\beta^{\mathrm{m}}(\mathbf{t}^{(1,2)})$. This inequality can be rewritten as: $\frac{E}{TP} > \frac{E+\Delta E_1+\Delta E_2}{TP+\Delta TP_1+\Delta TP_2}$. Taking into account also the assumptions of (B.2) we obtain:

$$\frac{E + \Delta E_1 + \Delta E_2}{TP + \Delta TP_1 + \Delta TP_2} < \frac{E}{TP} < \frac{E + \Delta E_k}{TP + \Delta TP_k}, k = 1, 2. \tag{B.3}$$

We consider two cases: $\Delta TP_2 < 0$, and $\Delta TP_2 > 0$ (the case $\Delta TP_2 = 0$ is trivial). If $\Delta TP_2 < 0$, applying (A.1) to the first and third term of (B.3),[1] we obtain $\frac{E+\Delta E_1}{TP+\Delta TP_1} < \frac{\Delta E_2}{\Delta TP_2}$. Using the second inequality of (B.3), we obtain $\frac{E}{TP} < \frac{\Delta E_2}{\Delta TP_2}$. Finally, applying (A.1) to the previous inequality,[2] we have $\frac{E+\Delta E_2}{TP+\Delta TP_2} < \frac{E}{TP}$, which contradicts the second inequality of (B.3) for $k = 2$.

The proof for the case $\Delta TP_2 > 0$ is similar. It can be obtained by applying (A.2) to the first and third term of Eq. (B.3),[3] then using the first of the inequalities (B.3), and finally applying (A.3),[4] which leads to a contradiction.

Inductive step. Assuming that (B.2) holds for each $m \leq m^* < N$, we have to prove that it holds also for $m = m^* + 1$, namely:

$$\text{if } \forall k \leq m^* + 1 \frac{E}{TP} < \frac{E + \Delta E_k}{TP + \Delta TP_k}, \text{ then } \frac{E}{TP} < \frac{E + \sum_{k=1}^{m^*+1} \Delta E_k}{TP + \sum_{k=1}^{m^*+1} \Delta TP_k}. \tag{B.4}$$

The consequent of (B.4) can be rewritten as:

$$\frac{E}{TP} < \frac{E + \sum_{k=1}^{m^*} \Delta E_k + \Delta E_{m^*+1}}{TP + \sum_{k=1}^{m^*} \Delta TP_k + \Delta TP_{m^*+1}}. \tag{B.5}$$

By the assumption on (B.2), we know that $\frac{E}{TP} < \frac{E+\sum_{k=1}^{m^*} \Delta E_k}{TP+\sum_{k=1}^{m^*} \Delta TP_k}$. The last inequality, together with the antecedent of (B.4), for $k = m^* + 1$, implies that (B.5) is true: the proof coincides indeed with the one of the basis case above, with a simple change of notation. This completes the proof of Property 1.

Appendix C. Proof of Property 2

Using the above notation, the first assumption can be rewritten as:

$$\forall \Delta E_k, \Delta TP_k \ \ \frac{E}{TP} < \frac{E + \Delta E_k}{TP + \Delta TP_k}, \tag{C.1}$$

where E and TP correspond to the given $\mathbf{t}$, while ΔE_k and ΔTP_k denote their changes, attained by changing the value t_k in $\mathbf{t}$ to some other value. The second assumption can be rewritten as:

$$\exists \Delta E_i, \Delta TP_i, i \neq k, \ \ \frac{E}{TP} > \frac{E + \sum_{i \neq k} \Delta E_i}{TP + \sum_{i \neq k} \Delta TP_i}. \tag{C.2}$$

[1] with $A = E + \Delta E_1$, $B = TP + \Delta TP_1$, $\Delta A = \Delta E_2$ and $\Delta B = \Delta TP_2 < 0$

[2] with $A = E$, $B = TP$, $\Delta A = \Delta E_2$, $\Delta B = \Delta TP_2 < 0$

[3] with $A = E + \Delta E_1$, $B = TP + \Delta TP_1$, $\Delta A = \Delta E_2$ and $\Delta B = \Delta TP_2 > 0$

[4] with $A = E$, $B = TP$, $\Delta A = \Delta E_2$, $\Delta B = \Delta TP_2 > 0$

Property 2 can thus be rewritten as:

$$\forall \Delta E'_k, \Delta TP'_k \;\; \frac{E + \sum_{i\neq k} \Delta E_i}{TP + \sum_{i\neq k} \Delta TP_i} < \frac{E + \sum_{i\neq k} \Delta E_i + \Delta E'_k}{TP + \sum_{i\neq k} \Delta TP_i + \Delta TP'_k}. \tag{C.3}$$

Since ΔE_k and ΔTP_k correspond to *any* change of the k-th threshold, we can consider the particular change that leads to $\Delta E_k = \Delta E'_k$ and $\Delta TP_k = \Delta TP'_k$. Using a T_k value lower than t_k implies $\Delta TP'_k \geq 0$. If $\Delta TP'_k = 0$, then $\Delta E' < 0$, and Property 2 is trivially true. If $\Delta TP'_k > 0$, inequality (C.1) implies $\frac{E}{TP} < \frac{E+\Delta E'_k}{TP+\Delta TP'_k}$. Applying (A.2) we obtain[5] $\frac{E+\Delta E'_k}{TP+\Delta TP'_k} < \frac{\Delta E'_k}{\Delta TP'_k}$. Combining the last inequality with (C.1), (C.2) we obtain $\frac{E+\sum_{i\neq k} \Delta E_i}{TP+\sum_{i\neq k} \Delta TP_i} < \frac{\Delta E'_k}{\Delta TP'_k}$. Applying (A.3) to the last inequality[6] we obtain (C.3), which completes the proof.

Appendix D. Computational complexity of Algorithm 1

An upper bound for the number of iterations of the `for` loop carried out by Algorithm 1 can be obtained under the following conditions:

1. For each class k, the scores $s_k(\mathbf{x}_i)$, $i = 1, \ldots, n$, are all different.
2. In each repeat-until loop, only one threshold is updated.
3. When any $t_k \in [s_k(\mathbf{x}_{(i)}), s_k(\mathbf{x}_{(i+1)}))$ is updated, for any $1 \leq i < n$, it takes a value in $[s_k(\mathbf{x}_{(i+1)}), s_k(\mathbf{x}_{(i+2)}))$.
4. The solution is provided by $t_k \in [s_k(\mathbf{x}_{(n)}), s_k(\mathbf{x}_{(n+1)}))$, $k = 1, \ldots, N$.

This implies that the Nn iterations of the `repeat-until` loop are carried out, and that in each iteration, one few value of the previously updated threshold has to be evaluated. The number of iterations of the `for` loop is thus: Nn (first `repeat-until` loop) $+Nn-1$ (second one), $+Nn-2$ (third one), ..., $+1$ (last one) $= \sum_{j=1}^{Nn} j = \frac{1}{2}\left(N^2n^2 + Nn\right) = O(n^2N^2)$.

Appendix E. Proof of Property 3

Micro F measure. For a given β, let $\mathbf{t}^* = \arg\max_{\mathbf{T}} \hat{F}^{\mathrm{m}}_{\beta}$. Assume that some $\mathbf{t} \neq \mathbf{t}^*$ exists, such that $\hat{p}^{\mathrm{m}}(\mathbf{t}) \geq \hat{p}^{\mathrm{m}}(\mathbf{t}^*)$, and $\hat{r}^{\mathrm{m}}(\mathbf{t}) \geq \hat{r}^{\mathrm{m}}(\mathbf{t}^*)$. This implies that: $\frac{1}{\hat{p}^{\mathrm{m}}(\mathbf{t})} + \frac{\beta^2}{\hat{r}^{\mathrm{m}}(\mathbf{t})} < \frac{1}{\hat{p}^{\mathrm{m}}(\mathbf{t}^*)} + \frac{\beta^2}{\hat{r}^{\mathrm{m}}(\mathbf{t}^*)}$, which (using Eq. (6)) implies in turn: $\hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t}) \geq \hat{F}^{\mathrm{m}}_{\beta}(\mathbf{t}^*)$. This contradicts the assumption $\mathbf{t}^* = \arg\max_{\mathbf{T}} \hat{F}^{\mathrm{m}}_{\beta}$.

Macro F measure. We prove Property 3 by constructing an example, in which the P-R point obtained by maximising $\hat{F}^{\mathrm{M}}_{\beta}$ does not belong to the macro-P-R curve. Consider a problem with two classes ($N = 2$), and assume that the P-R points A_1 and A_2 in Fig. F.5 (left) are the ones that maximise respectively $\hat{F}_{\beta,1}$ and $\hat{F}_{\beta,2}$, for some given β. This means that all other points of the P-R curves of these classes lie *below* the corresponding level curves of $\hat{F}_{\beta,k}$, $k = 1, 2$, that runs through A_1 and A_2 (shown as dashed lines). Since p^{M} and r^{M} are obtained by averaging p_k and r_k, $k = 1, 2$, the maximum of $\hat{F}^{\mathrm{M}}_{\beta}$ is attained by the midpoint of the segment joining A_1 and A_2, denoted in Fig. F.5 (left) with A. Consider now two other points of the P-R curves of the

[5] with $A = E$, $B = TP$, $\Delta A = \Delta E'_k$ and $\Delta B = \Delta TP'_k$ (we remind that $\Delta TP'_k > 0$).

[6] with $A = E + \sum_{i\neq k} \Delta E_i$, $B = TP + \sum_{i\neq k} \Delta TP_i$, $\Delta E'_k$ and $\Delta B = \Delta TP'_k$

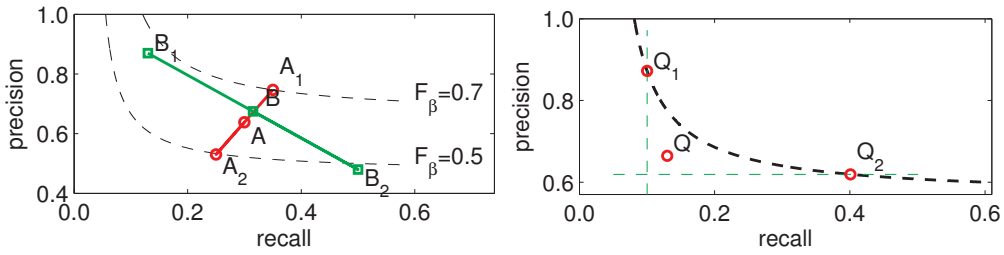

Figure F.5: Left: an example showing that maximising F_β^{M} can provide a point that does not belong to the macro-P-R curve (see Appendix E). Right: an example showing that a point Q of the micro-P-R curve can not be found by maximising F_β^{m}, if it lies below the level curve of F_β^{m} running through two other points of such curve (see Appendix F).

two classes, denoted respectively as B_1 and B_2. By the above assumption, they must lie below the corresponding level curves of $\hat{F}_{\beta,k}$, $k = 1, 2$, and thus they lead to a lower $\hat{F}_\beta^{\mathrm{M}}$ value than point A. However, it is easy to see that the corresponding macro-averaged point (denoted with B) dominates A. This means that A does not belong the macro-P-R curve, although it is obtained by maximising $\hat{F}_\beta^{\mathrm{M}}$.

Appendix F. Proof of property 4

Recall that a point Q of the P-R curve can be found by maximising the corresponding F measure, if at least one β values exists such that $\hat{F}_\beta(Q) > \hat{F}_\beta(Q')$, for each other point Q' of the P-R curve.

Micro F measure. Consider then three points of the micro-P-R curve, $Q = (r, p)$, $Q_1 = (r_1, p_1)$ and $Q_2 = (r_2, p_2)$, such that $r_1 < r < r_2$ and $p_1 > p > p_2$ (see Fig. F.5, right). We prove Property 4 by showing that, if and only if Q lies below the level curve of F_β passing through Q_1 and Q_2 (such level curve is unique, as shown below), no β value exists such that $\hat{F}_\beta^{\mathrm{m}}(Q) > \hat{F}_\beta^{\mathrm{m}}(Q_i)$, $i = 1, 2$.

Using Eq. (6), the inequalities above can be rewritten as: $\frac{1}{p} + \frac{\beta^2}{r} < \frac{1}{p_1} + \frac{\beta^2}{r_1}$, $\frac{1}{p} + \frac{\beta^2}{r} < \frac{1}{p_2} + \frac{\beta^2}{r_2}$. After some algebra we obtain: $\frac{1}{p} - \frac{1}{p_1} < \beta^2 \left(\frac{1}{r_1} - \frac{1}{r}\right)$, $\beta^2 \left(\frac{1}{r} - \frac{1}{r_2}\right) < \frac{1}{p_2} - \frac{1}{p}$. This easily leads to: $\frac{1/p - 1/p_1}{1/r_1 - 1/r} < \beta^2 < \frac{1/p_2 - 1/p}{1/r - 1/r_2}$. This means that the β values such that $\hat{F}_\beta^{\mathrm{m}}(Q) > \hat{F}_\beta^{\mathrm{m}}(Q_i)$, $i = 1, 2$, lie in the interval $\left(\frac{1/p - 1/p_1}{1/r_1 - 1/r}, \frac{1/p_2 - 1/p}{1/r - 1/r_2}\right)$. Consider now the level curve of the micro F passing through Q_1 and Q_2. The corresponding β' must satisfy: $\hat{F}_{\beta'}^{\mathrm{m}}(Q_1) = \hat{F}_{\beta'}^{\mathrm{m}}(Q_2)$. Rewriting this equality using Eq. (6), we obtain: $(\beta')^2 = \frac{1/p_2 - 1/p_1}{1/r_1 - 1/r_2}$. If Q lies below the level curve mentioned above, we have:

$$\hat{F}_{\beta'}^{\mathrm{m}}(Q) < \hat{F}_{\beta'}^{\mathrm{m}}(Q_1) = \hat{F}_{\beta'}^{\mathrm{m}}(Q_2)\,. \tag{F.1}$$

Using Eq. (6) with the above expression of β', the inequality in (F.1) related to Q_1 can be rewritten as: $\frac{1}{p} + \frac{1}{r}\left(\frac{1/p_2 - 1/p_1}{1/r_1 - 1/r_2}\right) > \frac{1}{p_1} + \frac{1}{r_1}\left(\frac{1/p_2 - 1/p_1}{1/r_1 - 1/r_2}\right)$, which after some algebra leads to: $\frac{1/p_2 - 1/p_1}{1/r_1 - 1/r_2} < \frac{1/p - 1/p_1}{1/r_1 - 1/r}$. Similarly, the inequality (F.1) related to Q_2 implies: $\frac{1/p_2 - 1/p_1}{1/r_1 - 1/r_2} > \frac{1/p_2 - 1/p}{1/r - 1/r_2}$. From the last two inequalities we finally obtain: $\frac{1/p - 1/p_1}{1/r_1 - 1/r} > \frac{1/p_2 - 1/p}{1/r - 1/r_2}$, which implies that the interval of β values defined above is empty.

Macro F measure. Note first that the point $(p^{\mathrm{M}}, r^{\mathrm{M}})$ obtained by maximising $\hat{F}_\beta^{\mathrm{M}}$ is the centroid of the points (p_k, r_k) obtained by maximising the corresponding $\hat{F}_{\beta,k}$, $k = 1, \ldots, N$.

Note also that the relationship between F_β^{m} and the point $(p^{\mathrm{m}}, r^{\mathrm{m}})$ is the same as between $F_{\beta,k}$ and (p_k, r_k). Together with the above proof, this implies that maximising each $\hat{F}_{\beta,k}$ does not guarantee to provide all the points of the P-R curve of each class and, therefore, of the macro-P-R curve.

Appendix G. Proof of Property 5

Micro F measure. Given any $\beta_1 \neq \beta_2$, the points $Q_1 = (r_1, p_1)$ and $Q_2 = (r_2, p_2)$ that maximise respectively $\hat{F}_{\beta_1}$ and $\hat{F}_{\beta_2}$ must satisfy the following inequalities: $F_{\beta_1}(Q_1) \geq F_{\beta_1}(Q_2)$ and $F_{\beta_2}(Q_2) \geq F_{\beta_2}(Q_1)$.

Using Eq. (6), the above inequalities can be rewritten respectively as: $(1/p_1 + \beta_1^2/r_1) \leq (1/p_2 + \beta_1^2/r_2)$, and $(1/p_2 + \beta_2^2/r_2) \leq (1/p_1 + \beta_2^2/r_1)$, which leads to:

$$\frac{1}{p_1} - \frac{1}{p_2} \leq {\beta_1}^2 \left(\frac{1}{r_2} - \frac{1}{r_1}\right), \;\; {\beta_2}^2 \left(\frac{1}{r_2} - \frac{1}{r_1}\right) \leq \frac{1}{p_1} - \frac{1}{p_2}. \tag{G.1}$$

Since the left-hand side (LHS) of the first inequality equals the right-hand side (RHS) of the second one, one obtains: ${\beta_2}^2(1/r_2 - 1/r_1) \leq {\beta_1}^2(1/r_2 - 1/r_1)$. After dividing both inequalities in (G.1) respectively by ${\beta_1}^2$ and ${\beta_2}^2$, the RHS of the former becomes identical to the LHS of the latter, which leads to: $1/{\beta_1}^2(1/p_1 - 1/p_2) \leq 1/{\beta_2}^2(1/p_1 - 1/p_2)$. The last two inequalities can be rewritten as:

$$\left({\beta_1}^2 - {\beta_2}^2\right)\left(\frac{1}{r_2} - \frac{1}{r_1}\right) \geq 0, \;\; \left(\frac{1}{{\beta_2}^2} - \frac{1}{{\beta_1}^2}\right)\left(\frac{1}{p_1} - \frac{1}{p_2}\right) \geq 0 \,.$$

If $\beta_1 \geq \beta_2$, the two inequalities above imply that: $(1/p_1 - 1/p_2) \geq 0$ and $(1/r_2 - 1/r_1) \geq 0$, which finally leads to $p_1 \leq p_2$, and $r_1 \geq r_2$.

Macro F measure. For the reasons explained at the end of Appendix F, Property 5 holds also for the $F_{\beta,k}$, p_k and r_k measures, $k = 1, \ldots, N$. Since the p^{M} and r^{M} values obtained by maximising $\hat{F}_\beta^{\mathrm{M}}$ are the average of the p_k and r_k obtained by maximising the corresponding $\hat{F}_{\beta,k}$ measures, it immediately follows that Property 5 holds also for F_β^{M}.

References

[1] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys 34 (1) (2002) 1–47.

[2] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, Data Mining and Knowledge Discovery Handbook (2010) 667–685.

[3] M. R. Boutell, J. Luo, X. Shen, C. M. Brown, Learning multi-label scene classification, Pattern Recognition 37 (9) (2004) 1757–1771.

[4] Y. Yang, A study on thresholding strategies for text categorization, in: Proc. Int. ACM SIGIR Conf. Research and Development in Information Retrieval, ACM, 2001, pp. 137–145.

[5] R.-E. Fan, C.-J. Lin, A study on threshold selection for multi-label, Tech. Rep., National Taiwan University (2007).

[6] C. J. van Rijsbergen, Information Retrieval, 2nd Edition, Butterworths, London, 1979.

[7] I. Pillai, G. Fumera, F. Roli, F-Measure Optimisation in Multi-label Classifiers, in: Proc. Int. Conf. Pattern Recognition (ICPR 2012), in press.

[8] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, Cambridge, 2008.

[9] S. Merler et al., Tuning cost-sensitive boosting and its application to melanoma diagnosis, in: Proc. Int. Workshop Multiple Classifier Systems, Springer LNCS Vol. 2096, 2001, pp. 32–42.

[10] D. D. Lewis et al., Rcv1: A new benchmark collection for text categorization research, J. Machine Learning Research 5 (2004) 361–397.

[11] W. R. Hersh et al.: An interactive retrieval evaluation and new large test collection for research, in: Proc. Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 1994, pp. 192–201.
[12] D. D. Lewis, R. E. Schapire, J. P. Callan, R. Papka, Training algorithms for linear text classifiers, in: Proc.Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 1996, pp. 298–306.
[13] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: Advances in Neural Inf. Proc. Systems 14, 2002, pp. 681–687.
[14] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, `http://www.csie.ntu.edu.tw/~cjlin/libsvm` (2001).
[15] Y. Yang, X. Liu, A re-examination of text categorization methods, in: Proc. Int. ACM SIGIR Conf. Research and Development in Information Retrieval, 1999, pp. 42–49.
[16] A. McCallum, K. Nigam, A comparison of event models for naive bayes text classification, in: Proc. AAAI Workshop on Learning for Text Categorization, Vol. 752, AAAI Press, 1998, pp. 41–48.
[17] Y. Yang, G. I. Webb, A comparative study of discretization methods for naive-bayes classifiers, in: Proc. Pacific Rim Knowledge Acquisition Workshop, 2002, pp. 159–173.
[18] I. Pillai, G. Fumera, F. Roli, A classification approach with a reject option for multi-label problems., in: Proc. Int. Conf. Image Analysis and Processing, Springer LNCS Vol. 6978, 2011, pp. 98–107.
[19] D. R. Musicant, V. Kumar, A. Ozgur, Optimizing f-measure with support vector machines, in: Proc. Int. Florida Artificial Intelligence Research Society Conference, AAAI Press, 2003, pp. 356–360.
[20] J. Davis, M. Goadrich, The relationship between precision-recall and ROC curves, in: Proc. Int. Conf. Mach. Learn., 2006, pp. 233–240.