Pattern Recognition 46 (2013) 2256–2266

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Multi-label classification with a reject option

Ignazio Pillai *, Giorgio Fumera, Fabio Roli

Department of Electrical and Electronic Engineering, University of Cagliari, Piazza d'Armi, 09123 Cagliari, Italy

ARTICLE INFO

Article history:
Received 26 June 2012
Received in revised form
22 January 2013
Accepted 27 January 2013
Available online 10 February 2013

Keywords:
Multi-label classification
Manual annotation
Reject option

ABSTRACT

We consider multi-label classification problems in application scenarios where classifier accuracy is not satisfactory, but manual annotation is too costly. In single-label problems, a well known solution consists of using a reject option, i.e., allowing a classifier to withhold unreliable decisions, leaving them (and only them) to human operators. We argue that this solution can be exploited also in multi-label problems. However, the current theoretical framework for classification with a reject option applies only to single-label problems. We thus develop a specific framework for multi-label ones. In particular, we extend multi-label accuracy measures to take into account rejections, and define manual annotation cost as a cost function. We then formalise the goal of attaining a desired trade-off between classifier accuracy on non-rejected decisions, and the cost of manually handling rejected decisions, as a constrained optimisation problem. We finally develop two possible implementations of our framework, tailored to the widely used *F* accuracy measure, and to the only cost models proposed so far for multi-label annotation tasks, and experimentally evaluate them on five application domains.

1. Introduction

A huge amount of text documents, images, videos and other kinds of multimedia data is currently available in digital form. Annotating them with semantic labels is necessary for their effective management and retrieval. Manual annotation is the traditional approach, but is infeasible for large amounts of data [1,2]. Accordingly, automatic annotation techniques have been the subject of a considerable research effort over the past ten years in the machine learning and pattern recognition communities [3–6]. Their accuracy is however not satisfactory in several real applications (see, e.g., [1] for image annotation, and [2,7] for text annotation). In such case, automatic annotation tools can be used only as a support for human annotators, who remain responsible of the final decisions (see, e.g., [1,7]).

In pattern recognition applications, a well known solution to attain a trade-off between classifier accuracy and cost of manual labelling is to use a *reject option*, i.e., allowing a classifier to withhold decisions deemed unreliable, leaving them (and only them) to human operators. In particular, this can be useful when the cost of misclassifications is higher than the cost of manual labelling [8,9]. We argue that a reject option may be useful also in annotation tasks, to attain a trade-off between the accuracy of automatic annotation and the cost (time) of manual annotation, in the case when: (i) manual annotation is too costly; (ii) automatic annotation techniques are not accurate enough; (iii) a certain amount of annotation errors is nevertheless tolerated (e.g., due to subjectiveness). However, the use of a reject option in annotation tasks has not been considered in the literature yet, except for our preliminary works [10–12]. Moreover, its formalisation in this context raises some theoretical issues. To this aim, the classical framework of [8,9] cannot be applied, since it refers to *single-label* (SL) problems only, and, in particular, to performance measures defined as the expected cost of the outcome of classifier decisions, including rejections. Annotation tasks are *multi-label* (ML) problems instead, and accuracy measures (e.g., precision and recall) are not defined as the expected cost of the outcome of classifier decisions. Note that in [13] a reject option based on a "multi-label classification rule" was proposed, but is not related to ML classification as intended in this work. Indeed, SL classifiers with loss function given by the expected cost were considered in [13]. The term "multi-label" was used to denote the fact that the proposed reject option was implemented by allowing a SL classifier to output more than one class label, in case of uncertainty about the true one; in this case, a human operator has to choose *one* of these labels. In our setting, samples can be assigned to more than one class; a ML classifier can withhold the decision about one or more classes, and each rejected decision is taken by a human operator.

Motivated by the above premises, that are discussed in detail in Section 2, in this paper we develop a specific framework for ML classification with a reject option (Section 3). We first extend ML accuracy measures to take into account only non-rejected

* Corresponding author. Tel.: +39 070 675 5817; fax: +39 070 675 5782.
E-mail addresses: pillai@diee.unica.it (I. Pillai), fumera@diee.unica.it (G. Fumera), roli@diee.unica.it (F. Roli).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.01.035

decisions about the relevance of the considered classes to a given sample, and formalise manual annotation cost as a cost function. This allows us to formalise the goal of attaining a trade-off between the two heterogeneous measures of classifier accuracy and manual annotation cost of rejections, as a constrained optimisation problem, which plays the role of the classifier learning problem. We also address two practical issues related to classifier design: how to define a suitable ML decision function with a reject option, and how to make the corresponding learning problem tractable. We then develop in Section 4 two possible implementations of our framework, tailored to the widely used F accuracy measure, and to two cost models proposed in [1] for image annotation, which are the only models formalised for ML problems so far. Our implementations are experimentally evaluated in Section 5 on eight benchmark data sets related to text, video, image, gene and music annotation tasks. The contributions of this work and directions for future research are finally discussed in Section 6.

2. Background

In this section we describe the framework for SL classification with a reject option of [8,9], and give an overview on ML classification.

2.1. Single-label classification problems with a reject option

In SL classification problems each sample belongs to a single class, out of N predefined ones. We denote with $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^n$ the feature vector of a given sample in a n-dimensional feature space $\mathcal{X}$, and with $y \in \mathcal{Y} = \{1,\ldots,N\}$ the corresponding class label. A classifier implements a decision function $f : \mathcal{X} \to \mathcal{Y}$. In statistical pattern recognition, SL problems have been formalised under the minimum risk framework, in which the outcomes of classifier decisions incur a cost defined by a loss function $L(y,f(\mathbf{x}))$. Classifier performance is measured as the expected risk $\mathbb{E}[L(Y,f(\mathbf{X})]$ (uppercase letters denote random variables). The simplest loss function is given by $L(y,f(\mathbf{x})) = \mathbb{I}[f(\mathbf{x}) = y]$, where $\mathbb{I}[a] = 1$ (0), if a=True (False). The corresponding $\mathbb{E}[L(Y,f(\mathbf{X})]$ equals the misclassification probability $\mathbb{P}[f(\mathbf{X}) \neq Y]$, which is minimised by assigning a sample to the class exhibiting the highest posterior (Bayes rule): $f(\mathbf{x}) = \arg\max_{k \in \mathcal{Y}} \mathbb{P}[Y = k|\mathbf{x}]$.

In applications where misclassifications are more costly than manual labelling, they can be reduced using a reject option, i.e., allowing the classifier to withhold uncertain decisions, and leaving them to human operators. A rejection can be conveniently represented as a fictitious class label 0: the decision function becomes $f : \mathcal{X} \to \{0\} \bigcup \mathcal{Y}$, and the loss function is extended to include the cost of manually handling rejections, $L(y,0)$ [8]. Note that, under this setting, classifier performance can be still measured as the expected risk. The simplest loss function with a reject option assigns the same cost to any rejection, $L(y,0) = \lambda_R \in (0,1)$. The corresponding expected risk is a linear combination of misclassification and rejection probabilities: $\mathbb{P}[f(\mathbf{X}) \neq Y,0] + \lambda_R \mathbb{P}[f(\mathbf{X}) = 0]$ [9]. The optimal decision rule is an extension of Bayes' rule: it rejects samples whose maximum posterior is below a threshold equal to $1-\lambda_R$ (Chow's rule). In this case, classifier performance can also be evaluated through the error-rejection curve, i.e., the functional relation between misclassification and rejection probabilities provided by Chow's rule for all possible λ_R values.

2.2. Multi-label classification

In ML problems each sample can belong to more than one class. We will denote the class labels of a sample as $\mathbf{y} = (y_1,\ldots,y_N) \in \{-1,+1\}^N$, where $y_k = +1$ (-1) means that $\mathbf{x}$ (does not) belong to class k. The decision function of a ML classifier has thus the form $f : \mathcal{X} \to \{-1,+1\}^N$.

Accuracy measures. ML problems usually occur in retrieval tasks, where performance is often measured as the probability that a retrieved sample is relevant to a query (precision), and the probability to retrieve a relevant sample (recall). In a ML classification problem, each class can be viewed as the set of samples that are relevant to a distinct query. Accordingly, the label-wise definition of precision and recall for class k, and the corresponding empirical estimates from a labelled data set, are defined as:

$$p_k = \mathbb{P}[Y_k = +1 | f_k(\mathbf{X}) = +1], \quad r_k = \mathbb{P}[f_k(\mathbf{X}) = +1 | Y_k = +1], \tag{1}$$

$$\hat{p}_k = TP_k/(TP_k + FP_k), \quad \hat{r}_k = TP_k/(TP_k + FN_k), \tag{2}$$

where TP_k, FP_k and FN_k denote respectively the number of true positive, false positive and false negative samples. A widely used scalar combination of p_k and r_k is van Rijsbergen's F measure

$$F_{\beta,k} = \frac{1+\beta^2}{1/p_k + \beta^2/r_k} \in (0,1], \tag{3}$$

where $\beta \in [0,+\infty]$ allows one to give a different weight to p_k and r_k. The above measures can also be defined sample-wise, by viewing a class as the set of queries that are relevant to a given sample [5]. Other sample-wise measures also exist, like Hamming loss and ranking loss [5].

For label- and sample-wise measures, the overall accuracy is empirically defined by averaging respectively over classes and samples ("macro-averaging"). In particular, the label-wise macro F measure is given by

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

The overall accuracy can also be empirically measured by "micro-averaging", i.e., by considering all predictions over labels and samples simultaneously in the computation of precision and recall of Eq. (2) [3,5]. This makes micro-averaged measures usually more difficult to maximise than macro-averaged ones. In particular, the micro F measure is defined as

$$\hat{F}_\beta^{\mathrm{m}} = \left(1 + \frac{1}{1+\beta^2}\frac{\sum_{k=1}^{N}(FP_k + \beta^2 FN_k)}{\sum_{k=1}^{N} TP_k}\right)^{-1}. \tag{5}$$

Manual annotation cost. Manual annotation cost can be measured in terms of annotation time. For a given sample, it may depend on several application-specific factors, besides the number N of decisions to take. For instance: the number of 'relevant' and 'non-relevant' decisions, the time needed to analyse a sample, the specific annotation technique, the correlation between labels (deciding whether labelling or not a sample as belonging to two or more correlated classes may require a lower time than for independent classes), and class frequency (deciding for rare classes may require a higher time than for common ones). To our knowledge, no general cost model has been developed so far for ML problems, and only two specific models have been proposed, for the *tagging* and *browsing* image annotation techniques [1]. We summarise them in the following, since we will exploit them in the rest of this paper.

Tagging consists of annotating one image at a time, with respect to all the classes. It was modelled in [1] by assuming that an image is first analysed for an average "setup" time t_s, and that an average time t_f is then spent for assigning (e.g., typing, or selecting) the labels of each relevant class, while deciding about non-relevant classes requires a negligible time. Browsing consists instead of annotating a set of images, with respect to a given class. It was modelled by assuming that an average time t_p and t_n is

spent to decide respectively whether each image is relevant or not to the given class, with $t_n < t_p$. In both cases, an additional zero-mean noise term ϵ is considered. Note that these models do not take into account the possible effects of class frequency and correlation. Denoting with $N_p(\mathbf{x})$ the number of classes relevant to image $\mathbf{x}$, the respective annotation times are given by

$$t_t = t_s + N_p(\mathbf{x})t_f + \epsilon, \quad t_b = N_p(\mathbf{x})t_p + [N - N_p(\mathbf{x})]t_n + \epsilon. \tag{6}$$

Accuracy–cost trade-off. In SL problems classification accuracy and cost of rejections are *homogeneous* quantities, defined in terms of costs of classification outcomes. Their trade-off can thus be evaluated using a *single* measure, the expected risk (cost). In ML problems these measures are *heterogeneous* instead, since accuracy is not associated to costs of classification outcomes. This implies that, if a reject option is used, their trade-off has to be evaluated by considering them *separately*.

3. A framework for multi-label classification with a reject option

In this section we formalise the problem of designing a ML classifier with a reject option, by defining the form of the decision function and the measures of classification accuracy and cost of rejections. We then define the corresponding learning problem, and discuss some implementation issues.

3.1. Decision function and performance measure

Decision function. We consider the most general kind of ML decision function with a reject option: it allows a classifier to withhold any subset of the N decisions (including none, and all of them) whether labelling a given sample as belonging or not to the corresponding classes. Denoting a rejection decision with the label '0', such kind of decision function has the form

$$f : \mathcal{X} \to \{-1, 0, +1\}^N. \tag{7}$$

Classification accuracy. When a reject option is used, accuracy must be evaluated over non-rejected decisions only. To this aim, existing ML accuracy measures have to be extended. In this work we focus on the two most widely used measures, i.e., the label-wise macro F, and the micro F (see Section 2.2). Other label- and sample-wise measures can be extended similarly.

We start from precision and recall of a single class (Eq. (1)). By analogy with SL problems, in which accuracy is defined as the conditional probability that a sample is correctly classified, given that it has not been rejected [9], we extend the definition of precision and recall by conditioning the corresponding probabilities to $f_k(\mathbf{x}) \neq 0$ (note that the original definition of precision is implicitly conditioned to $f_k(\mathbf{x}) \neq 0$, and thus remains unchanged)

$$p_k = \mathbb{P}[Y_k = +1 \,|\, f_k(\mathbf{X}) = +1],$$
$$r_k = \mathbb{P}[f_k(\mathbf{X}) = +1 \,|\, Y_k = +1, f_k(\mathbf{X}) \neq 0].$$

The empirical estimates $\hat{p}_k$ and $\hat{r}_k$ can still be obtained as Eq. (2); however, the values TP_k, FP_k, FN_k and TN_k must be computed over non-rejected decisions only, according to the contingency table reported in Table 1. Similarly, the F measure (either for a class, or a sample), as well as the macro (both label- and sample-wise) and micro precision, recall and F measures, can be computed as in Section 2.2, using the above contingency table.

Manual annotation cost. It can be formalised, similarly to SL problems, through a cost function $C(\mathbf{y}, f(\mathbf{x}))$. In this case, it is defined as the time needed to annotate $\mathbf{x}$, considering only the classes k whose decision has been rejected, i.e., $f_k(\mathbf{x}) = 0$. The exact expression is clearly application-specific. For instance, the cost function corresponding to the tagging and browsing models for image annotation (Eq. (6)) is given by, respectively

$$C_t(\mathbf{y}, f(\mathbf{x})) = t_s + t_f \sum_{k=1}^{N} \mathbb{I}[y_k = +1, f_k(\mathbf{x}) = 0],$$

$$C_b(\mathbf{y}, f(\mathbf{x})) = \sum_{k=1}^{N} (t_p \mathbb{I}[y_k = +1, f_k(\mathbf{x}) = 0] + t_n \mathbb{I}[y_k = -1, f_k(\mathbf{x}) = 0]). \tag{8}$$

Table 1
Contingency table for class k, for a ML classifier with a reject option. RR_k and NR_k denote respectively rejected decisions for *R*elevant and *N*on-relevant samples.

$f_k(x)$		$+1$	0	-1
y_k	$+1$	TP_k	RR_k	FN_k
	-1	FP_k	NR_k	TN_k

3.2. Classifier learning problem

Let $f(\cdot;\theta)$ be the chosen decision function, where θ is a set of parameters to be set by a learning algorithm, and $A(\theta)$ the corresponding value of the chosen accuracy measure, on non-rejected decisions. The latter can be increased at the expense of a higher amount of rejections, i.e., a higher manual annotation cost, similarly to SL problems [9]. It is thus necessary to find a trade-off between them, according to application requirements. An interesting kind of application requirement, which is often used also in SL problems, is to maximise accuracy, with the constraint that the expected manual annotation cost does not exceed a given value C_{max}. This can be formalised as a constrained optimisation problem, which plays the role of the classifier learning problem, in terms of empirical estimates of accuracy and cost on a given training set of M samples

$$\max_{\theta} \hat{A}(\theta) \quad \text{s.t. } \frac{1}{M} \sum_{i=1}^{M} C(\mathbf{y}_i, f(\mathbf{x}_i; \theta)) \leq C_{max}. \tag{9}$$

Two other kinds of application requirements can be expressed, in terms of accuracy and cost: (i) minimising the expected cost, with the constraint that accuracy is not below a given value A_{min}; (ii) attaining an accuracy not lower than A_{min} and an expected cost not higher than C_{max}. We do not discuss them in the rest of this paper, as they can be dealt with by solving problem (9) (see Section 4).

Consider finally the choice of $f(\cdot;\theta)$. Note that in SL problems the optimal decision rule with a reject option (Chow's rule) is analytically known, in terms of posterior probabilities (see Section 2.1). One can thus use the plug-in principle to define a decision function in practice (when the exact posteriors are unknown), i.e., applying Chow's rule to posteriors' estimates. A similar solution could be used also for ML problems. However, for some ML accuracy measures, it may be not possible to obtain the optimal solution of problem (9) analytically. In particular, in Appendix A (online supplementary material) we show that this is the case of the micro and macro F measures. In such cases, only heuristic choices of $f(\cdot;\theta)$ can be made. A possible criterion is proposed in the next section.

3.3. Implementation issues

Here we discuss issues related to the choice of a decision function $f(\cdot;\theta)$, and to the development of optimisation algorithms to solve problem (9).

First, a decision function with a reject option can be defined in two different ways. One is to define a function $f(\cdot;\theta)$ that directly

maps form feature space $\mathcal{X}$ to decision space $\{-1,0,1\}^N$. An alternative approach is to first training a classifier without a reject option, $g:\mathcal{X}\rightarrow\{-1,1\}^N$ (or $g:\mathcal{X}\rightarrow\mathbb{R}^N$, for classifiers that provide real-valued scores), and then defining a decision function $f(\cdot;\theta)$ that maps from the outputs of $g(\cdot)$ to $\{-1,0,1\}^N$. The latter approach is widely used in SL problems (see, e.g., [14,15]).

Consider now the issue of developing an optimisation algorithm for solving problem (9). Depending on the cost and decision functions, and on the accuracy measure, it can be very difficult to find an effective and efficient optimisation algorithm, and (9) may even be computationally intractable. A zero-effort solution is to resort to general-purpose, although suboptimal, algorithms (e.g., genetic algorithms). Another possibility is to choose a function $f(\cdot;\theta)$ which makes it easier to develop a specific algorithm (possibly optimal, if any) for solving problem (9). However, this may be not sufficient. For instance, we were not able to find any such $f(\cdot;\theta)$, in the specific cases when $\hat{A}=\hat{F}^{\mathrm{M}}$ or $\hat{A}=\hat{F}^{\mathrm{m}}$, and either of the cost functions (8) is used. In such a case, a further possibility is to first define a proper approximation of the cost function at hand, and then to choose a suitable $f(\cdot;\theta)$. Some examples of the latter strategy are given in the next section.

4. Implementation of multi-label classifiers with a reject option

In this section we propose a possible implementation of ML classifiers with a reject option, following the framework of Section 3. We focus on the widely used micro and (label-wise) macro F accuracy measures, and on the cost functions of Eq. (8), that correspond to the only formal cost models proposed so far for annotation tasks [1]. Note that, although these models refer to image annotation, they can be valid for other annotation tasks as well, if similar annotation techniques are used. With regard to the choice of the decision function, we will use the second approach mentioned in Section 3.3, i.e., defining a rejection criteria on the outputs of a trained ML classifier without a reject option. We start by defining suitable approximations of the considered cost models, and the corresponding decision functions, for the reasons explained in Section 3.3. We then develop the respective learning algorithms.

4.1. Approximation of cost models and choice of decision functions

Browsing cost model. The browsing cost function of Eq. (6) (right) can be approximated, when $t_p\approx t_n$, by setting $t_p=t_n=t_{\mathrm{d}}$, i.e., assuming that deciding about the relevance of any class to a sample requires a constant time t_{d}, independently on the actual relevance. This leads to

$$C(\mathbf{y},f(\mathbf{x};\theta))=t_{\mathrm{d}}\sum_{k=1}^{N}\mathbb{I}[f_k(\mathbf{x};\theta)=0]. \tag{10}$$

This implies that the overall manual annotation cost is proportional to the number of rejected decisions, regardless of *which* decisions are rejected, i.e., of how they are distributed across samples, and of the corresponding correct decisions (either 'relevant' or 'non-relevant'). This allows one to control the accuracy–cost trade-off by tuning the fraction of rejected decisions, that we call "rejection rate". Using cost function (10), we will show that learning problem (9) can be simplified by choosing a decision function $f(\cdot;\theta)$ that allows the classifier to reject the decision about each individual class, *independently* on the other classes. We call this kind of rejection strategy as "rejection of decisions".

Tagging cost model. The tagging cost function of Eq. (6) (left) can be approximated, when $t_{\mathrm{s}}>>N_{\mathrm{p}}(\mathbf{x})t_{\mathrm{f}}$, by assuming that manually handling a sample that contains rejected decisions requires a constant time t_{s}, regardless of the number of rejected decisions

$$C(\mathbf{y},f(\mathbf{x};\theta))=t_{\mathrm{s}}\mathbb{I}\left[\bigvee_{k=1}^{N}f_k(\mathbf{x};\theta)=0\right]. \tag{11}$$

This can be realistic in tasks in which most of the annotation time is spent for analysing a sample, and a relatively much lower time is needed to decide about the relevance of each class. The overall manual annotation cost is thus proportional to the number of samples for which at least one decision is rejected. Problem (9) can thus be simplified by choosing a decision function $f(\cdot;\theta)$ that allows the classifier to reject either *all* the N decisions for a sample, or *none* of them. The corresponding accuracy–cost trade-off can thus be controlled by choosing the fraction of samples that will be manually annotated. We will refer to them as "rejected samples", and to their fraction as "rejection rate". We call this kind of rejection strategy "rejection of samples".

4.2. Implementation under the approximated browsing cost model

To simplify problem (9) under the approximated browsing cost model (10), we proposed to define a decision function $f(\mathbf{x};\theta)$ that allows a classifier to reject the decision about each individual class, independently on the other classes. We discuss first how such a decision function can be defined.

Many ML classifiers output a real-valued score $s_k(\mathbf{x})\in\mathbb{R}$, $k=1,\ldots,N$, representing the "likelihood" that class k is relevant to $\mathbf{x}$ [16,17] (note that the scores are not necessarily calibrated probabilities, e.g., in the case of support vector machine classifiers). In the standard case without a reject option, the decision function $f(\cdot)$ is usually implemented by setting a threshold t_k for each class, such that $f_k(\mathbf{x})=+1(-1)$, if $s_k(\mathbf{x})\geq t_k(<t_k)$. The values of the N thresholds can be set by maximising accuracy on validation data, using the scores of the trained classifier (see, e.g., [16–18]). More complex thresholding strategies also exist, like the one of [19], which implements a decision function (without a reject option) for ML classifiers that output estimates of class posterior probabilities.

In these cases, the reliability of a decision $f_k(\mathbf{x})$ can be associated to the distance between the score $s_k(\mathbf{x})$ and the corresponding threshold t_k: intuitively, the higher $|s_k(\mathbf{x})-t_k|$, the higher the reliability. A reject option can thus be implemented by using a pair of thresholds for each class, $t_k^{\mathrm{L}}\leq t_k^{\mathrm{H}}$, $k=1,\ldots,N$ (where 'L' and 'H' stand respectively for "lower" and "higher"), such that

$$f_k(\mathbf{x};\theta)=\begin{cases}+1 & \text{if } s_k(\mathbf{x})\geq t_k^{\mathrm{H}},\\ 0 & \text{if } t_k^{\mathrm{L}}\leq s_k(\mathbf{x})<t_k^{\mathrm{H}},\\ -1 & \text{if } s_k(\mathbf{x})<t_k^{\mathrm{L}},\end{cases} \tag{12}$$

where $\theta=(t_1^{\mathrm{L}},t_1^{\mathrm{H}},\ldots,t_N^{\mathrm{L}},t_N^{\mathrm{H}})$. Note that this is the same kind of decision function with a reject option proposed in [15] for binary SL problems. The difference with our problem lies in the criterion that must be used for choosing threshold values, which is related to misclassification probability in [15], and to different ML accuracy measures in our case.

Using decision function (12), problem (9) amounts to find the values of the $2N$ thresholds, given the scores provided by a trained ML classifier on M validation samples. The constraint of problem (9) can be rewritten as follows, taking into account cost function (10)

$$\frac{1}{M}\sum_{i=1}^{M}\left(\sum_{k=1}^{N}\mathbb{I}[f_k^{\mathrm{R}}(\mathbf{x}_i;\theta)=0]\right)\leq\frac{C_{\max}}{t_{\mathrm{d}}}. \tag{13}$$

In words, the average number of rejected decisions per sample must not exceed $C_{\max}/t_d$. Denoting the rejection rate (i.e., the fraction of rejected decisions out of the MN overall decisions, see Section 4.1) with r, constraint (13) is equivalent to $r \le r_{\max} = C_{\max}/Nt_d$. To efficiently solve problem (9), it is convenient to set a distinct constraint for each class, by requiring that the rejection rate of class k, denoted as r_k, does not exceed a given value $r_{\max,k}$

$$r_k = \frac{1}{M}\sum_{i=1}^{M} \mathbb{I}[f_k^R(\mathbf{x}_i;\theta)=0] \le r_{\max,k}, \quad k=1,\ldots,N. \tag{14}$$

Choosing the values $r_{\max,k}$ such that $\sum_{k=1}^{N} r_{\max,k} = C_{\max}/t_d$, one obtains a stronger constraint than (13). Accordingly, to attain a rejection rate $r_{\max}$, one may need to (empirically) choose values of $r_{\max,k}$ such that $\sum_{k=1}^{N} r_{\max,k} > C_{\max}/t_d$. Using constraint (14), problem (9) becomes

$$\max_{\theta} \quad \hat{A}(\theta)$$
$$\text{s.t.} \quad r_k(\theta) \le r_{\max,k}, \quad t_k^L \le t_k^H, \ k=1,\ldots,N. \tag{15}$$

It is now easy to see that, when $\hat{A}(\theta) = \hat{F}_\beta^M(\theta)$ (see Eq. (3)), problem (15) can be decomposed into N independent problems, each one involving the $\hat{F}_{\beta,k}$ measure of a single class and the two corresponding thresholds

$$\max_{t_k^L, t_k^H} \quad \hat{F}_{\beta,k}(t_k^L, t_k^H)$$
$$\text{s.t.} \quad r_k(t_k^L, t_k^H) \le r_{\max,k}, \ \ t_k^L \le t_k^H. \tag{16}$$

In Appendix C of the online supplementary material we show that the optimal solution of problem (16) can be found at $O(M^2)$ computational cost.

The case when $\hat{A}(\theta) = \hat{F}_\beta^m(\theta)$ is more complex, since $\hat{F}_\beta^m$ cannot be decomposed over classes (see Eq. (5)), and exhaustive search is impractical. Nevertheless, we derived two properties of $\hat{F}_\beta^m(\theta)$ that allow the optimal solution of problem (15) to be found with low computational cost. These properties extend the ones we derived in [18] for the standard $\hat{F}_\beta^m$ measure without a reject option. Their proof, and the derivation of the corresponding optimisation procedure, reported here as Algorithm 1, can be found in the online Appendix C. Basically, Algorithm 1 iteratively scans the N pairs of thresholds, and updates each of them to the value that *locally* maximises $\hat{F}_\beta^m$, by keeping all the other $N-1$ threshold pairs at their current value. The algorithm stops when no increase of $\hat{F}_\beta^m$ is attained, after a whole scan of the N threshold pairs. In the online Appendix D we show that the computational complexity of Algorithm 1 is upper bounded by $O(N^2 M^3)$. In Section 5.2 we show that the actual computational cost can be much lower.

Algorithm 1 can also be exploited to approximate the optimal solution of learning problems defined according to alternative kinds of application requirements mentioned in Section 3.2. To this aim, one can solve problem (9) for different $C_{\max}$ values, and then choose the solution that provides the best accuracy–cost trade off according to the given application requirement.

Algorithm 1. Optimisation algorithm for problem (15), when $\hat{A} = \hat{F}_\beta^m$.

Input: a set of labelled samples $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^M$;
the scores $s_k(\mathbf{x}_i)$, $k=1,\ldots,N$, $i=1,\ldots,M$, provided by a ML classifier
Output: threshold values $t_1^L, t_1^H, \ldots, t_N^L, t_N^H$
  set $t_k^L = t_k^H = \min_{i=1,\ldots,M} s_k(\mathbf{x}_i)$, $k=1,\ldots,N$
  repeat
    updated ← False
    for $k=1, \ldots, N$ **do**
      $(t_k^{*L}, t_k^{*H}) \leftarrow \arg\max_{(\tau^L,\tau^H)\in\mathcal{T}} \hat{F}_\beta^m(t_1^L, t_1^H, \ldots, \tau^L, \tau^H, \ldots, t_N^L, t_N^H)$,
      where
$$\mathcal{T} = \{(\tau^L,\tau^H) \in \mathbb{R}^2 : \tau^H > t_k^H, \tau^H > \tau^L, r_k(\tau^L,\tau^H) \le r_{\max,k}\}$$
      if
$$\hat{F}_\beta^m(t_1^L, t_1^H, \ldots, t_k^{*L}, t_k^{*H}, \ldots, t_N^L, t_N^H) > \hat{F}_\beta^m(t_1^L, t_1^H, \ldots, t_k^L, t_k^H, \ldots, t_N^L, t_N^H)$$
      then $(t_k^L, t_k^H) \leftarrow (t_k^{*L}, t_k^{*H})$, *updated* ← True **end if**
    end for
  until *updated* = True
  return $(t_1^L, t_1^H, \ldots, t_N^L, t_N^H)$

4.3. Implementation under approximated tagging cost model

Under cost model (11), we proposed to define a decision function that allows the classifier to reject either all the N decisions for an input sample, or none of them. A possible solution is to first training any ML classifier without a reject option, and then defining a measure of classification reliability $R : \mathcal{X} \to \mathbb{R}$ based on its crisp outputs $g(\mathbf{x}) \in \{-1,+1\}^N$ (or soft outputs $s_k(\mathbf{x}) \in \mathbb{R}$, $k=1,\ldots,N$). Assuming that a higher $R(\mathbf{x})$ denotes a higher reliability, a rejection threshold t can be set, such that the resulting decision function $f(\mathbf{x};\theta)$, with $\theta = \{t\}$, is given by

$$f_k(\mathbf{x};t) = \begin{cases} g_k(\mathbf{x}) & \text{if } R(\mathbf{x}) \ge t \\ 0, & \text{otherwise} \end{cases} \quad k=1,\ldots,N. \tag{17}$$

Using cost function (11), the constraint of problem (9) becomes

$$\frac{1}{M}\sum_{i=1}^{M} (\mathbb{I}[R(\mathbf{x}_i) < t]) \le \frac{C_{\max}}{t_s}. \tag{18}$$

In words, the fraction of rejected samples (rejection rate) must not exceed $C_{\max}/t_s$. Denoting these two quantities respectively as r and $r_{\max}$, problem (9) can be rewritten as

$$\underset{t}{\text{maximise}} \quad \hat{A}(t)$$
$$\text{s.t.} \quad r(t) \le r_{\max}. \tag{19}$$

For any given $R(\cdot)$, the above problem could be solved analytically (depending on the accuracy and reliability measures), or by a simple iterative search over the possible t values. Optimisation problems corresponding to alternative application requirements mentioned in Section 3.2 can be dealt with by solving problem (19) for different $r_{\max}$ values, similarly to Section 4.2.

Consider now how to define $R(\cdot)$. Note that different $R(\cdot)$ can lead to different solutions of problem (19), being equal $r_{\max}$. Accordingly, denoting with $t_{R(\cdot)}$ the optimal solution of problem (19) for a given $R(\cdot)$, the "best" reliability measure $R^*(\cdot)$ is given by $R^*(\cdot) = \arg\max_{R(\cdot)} \hat{A}(t_{R(\cdot)})$. In words, $R^*(\cdot)$ is the reliability measure that allows one to obtain the highest accuracy on non-rejected samples, for any given $r_{\max}$. However, since the decision function that maximises $\hat{F}_\beta^M$ and $\hat{F}_\beta^m$ cannot be found analytically (see Section 3.2), the same holds for $R^*(\cdot)$, when $\hat{A} = \hat{F}_\beta^M$ or $\hat{A} = \hat{F}_\beta^m$. In particular, since $\hat{F}_\beta^M$ and $\hat{F}_\beta^m$ are not defined sample-wise, the contribution of a given sample $\mathbf{x}$ cannot be decoupled from the one of the other samples used to evaluate them. Consequently, only heuristic criteria can be used to define $R(\cdot)$.

To find a reasonable $R(\cdot)$, we analysed the expressions of $\hat{F}_\beta^M$ and $\hat{F}_\beta^m$ (see Eqs. (4) and (5)), to check whether some conditions on the TP, FN and FP values of a given set of samples exist, under which these measures become additive over samples. Our idea

was to derive the "best" reliability measure $R^*(\cdot)$ under such conditions, and use it as a (suboptimal) reliability measure for all samples. We were able to find such conditions for $\hat{F}_\beta^{\mathrm{m}}$, and the corresponding reliability measure turns out to be (see Appendix E of the online supplementary material)

$$R(\mathbf{x}) = \frac{TP(\{\mathbf{x}\}) + A}{FP(\{\mathbf{x}\}) + \beta^2 FN(\{\mathbf{x}\}) + B}, \tag{20}$$

where $TP(\{\mathbf{x}\})$ denotes the number of true positive decisions for $\mathbf{x}$, and similarly for $FP(\{\mathbf{x}\})$ and $FN(\{\mathbf{x}\})$, while A and B are *arbitrary* positive constants. We did not find any analogous condition for $\hat{F}_\beta^{\mathrm{M}}$, instead, but only for the $\hat{F}_{\beta,k}$ measure of each class. The corresponding reliability measure has a similar expression as Eq. (20): $(TP_k(\{\mathbf{x}\}) + A_k)/(FP_k(\{\mathbf{x}\}) + \beta^2 FN_k(\{\mathbf{x}\}) + B_k)$, where $TP_k(\{\mathbf{x}\}) \in \{0,1\}$ indicates whether $\mathbf{x}$ is a true positive (1) or not (0) for class k, and similarly for $FP_k(\{\mathbf{x}\})$ and $FN_k(\{\mathbf{x}\})$, while A_k and B_k are arbitrary positive constants as before (see online Appendix E). This suggests the following reliability measure for $\hat{F}_\beta^{\mathrm{M}}$, which mimicks the macro-averaging criterion:

$$R(\mathbf{x}) = \frac{1}{N}\sum_{k=1}^{N} \frac{TP_k(\{\mathbf{x}\}) + A_k}{FP_k(\{\mathbf{x}\}) + \beta^2 FN_k(\{\mathbf{x}\}) + B_k}. \tag{21}$$

A possible estimate of $TP(\{\mathbf{x}\})$, $FP(\{\mathbf{x}\})$, and $FN(\{\mathbf{x}\})$ in Eq. (20), and of the corresponding values in Eq. (21), is proposed in Section 5.3. The values of A and B, and of A_k and B_k, $k = 1, \ldots, N$, can be set using validation data, as shown in the online Appendix E.

5. Experiments

In this section we give an experimental evaluation of the two implementations of ML classifiers with a reject option developed in Section 4, in different annotation tasks: text, image, video, music, and gene annotation. Recall that our implementations are tailored to the cost models of image annotation with tagging and browsing. Accordingly, for data sets related to image annotation, our experiments refer to the case when the manual annotation techniques are tagging or browsing. For data sets related to other annotation tasks (for which no cost model has been proposed so far), our experiments can be considered representative of a scenario in which the underlying annotation techniques are similar to tagging and browsing, in the sense that the corresponding cost models can be approximated by (or, possibly, are exactly equal to) the ones we defined in Section 4.1.

5.1. Data sets and classifiers

We used nine benchmark ML data sets: Reuters 21578, the five subsets of Reuters RCV1v2 [20], the Heart Disease sub-tree of Ohsumed [21], and the Tmc2007 SIAM Text Mining Competition data set (text categorisation); Scene and Corel-5k (image annotation); Yeast (gene annotation); Mediamill (video annotation); Emotions (music annotation). 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 Corel-5k we used the feature vectors of [22] [1]; for Scene, Yeast, Mediamill, Emotions and RCV1v2, we used the feature vectors of [17].[2] For the other data sets we used tf–idf features, and carried out stemming, stop-word removal, and a further feature selection step using the information gain criterion [3]. The main characteristics of the data sets are reported in Table 2.

[1] http://mulan.sourceforge.net/datasets.html.
[2] http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html.

Table 2
Characteristics of the data sets used in the experiments. For RCV1v2, average values over the five available training sets are reported.

Dataset	Samples (training/testing)	Features	Classes	Class freq. (min/max)	Labels per sample (mean ± std. dev.)
Reuters	7769/3019	18 157	90	1E−4/0.37	1.23 ± 0.71
Ohsumed	12 775/3750	17341	99	2E−4/0.25	1.49 ± 0.87
RCV1v2	3000/3000	47 237	101	3E−4/0.46	3.19 ± 1.36
Tmc2007	21 519/7077	30438	22	0.01/0.60	2.23 ± 1.07
Yeast	1500/917	104	14	0.06/0.75	4.23 ± 1.58
Scene	1211/1196	295	6	0.14/0.23	1.06 ± 0.25
Mediamill	30 993/12 914	120	101	0.04/0.78	4.36 ± 2.30
Emotions	391/202	72	6	0.30/0.43	1.81 ± 0.67
Corel-5k	4500/500	499	374	2E−4/0.22	3.52 ± 0.66

We implemented ML classifiers using the well known binary relevance approach, i.e., by independently training a binary classifier for each class [3,5]. Although it disregards correlation between classes, contrary to more complex approaches (e.g., [23]), it is widely used due to its limited computational cost. We used two different statistical classifiers widely used in the ML literature: support vector machines (SVMs) [24], with linear kernel for data sets related to text categorisation, and RBF kernel for the other data sets, and k-nearest neighbours (k-NN) [25]. For data sets related to text categorisation we also used the ad hoc version of the Naive Bayes (NB) classifier of [26]. NB was not used for the other data sets, as well as k-NN for Mediamill and Corel-5k, due to their poor performance. Selection of features and of classifier parameters was carried out through a four-fold cross-validation (CV) on the original training set (the first training set was used for RCV1v2).

Ten runs of the experiments were carried out: the original training set was split into ten disjoint subsets, and eight of them were randomly chosen as the training set in each run. The original testing set was always used for performance evaluation. Since all the considered classifiers output a real-valued score for each category, we implemented decision functions without a reject option as described in Section 4.2, using a distinct threshold for each class. Such thresholds were computed by maximising the accuracy measure, either the macro or micro F, through a five-fold cross-validation on the training set of each run (the optimisation algorithm of [18] was used for micro F). Threshold values of decision functions with a reject option (either $t_1^{\mathrm{L}}, t_1^{\mathrm{H}}, \ldots, t_N^{\mathrm{L}}, t_N^{\mathrm{H}}$, or t) were computed through a similar cross-validation procedure, using the algorithms presented in the previous sections. In these experiments we considered the F_1 measure ($\beta = 1$) only. Its average values and standard deviation will be reported in the following, over the ten runs of the experiments.

5.2. Results for the approximated browsing cost model

Figs. 1 and 2 (first and second columns) report the accuracy–rejection curves attained on the nine data sets, for each considered base classifier, using the implementation of a reject option of Section 4.2. Recall that, under cost model (10), manual annotation cost of rejections is proportional to the fraction r of rejected decisions (rejection rate). Values of $r_{\max}$ between 0.0 and 0.3 were considered, with steps of 0.05. The maximum rejection rate per class (see Section 4.2) was set to $r_{\max,k} = r_{\max}$. The accuracy values for $r = 0$ are the ones of the standard ML classifiers without a reject option.

Figs. 1 and 2 (first column) show that the use of a reject option always provided an increase of $\hat{F}_1^{\mathrm{m}}$, for increasing rejection rates, which is the desired behaviour. For instance, using a SVM classifier on Reuters, the average $\hat{F}_1^{\mathrm{m}}$ increased from 0.87 to 0.97, by rejecting only 5% of decisions. Assuming that cost model (10) is exact, the corresponding manual annotation cost equals 5%

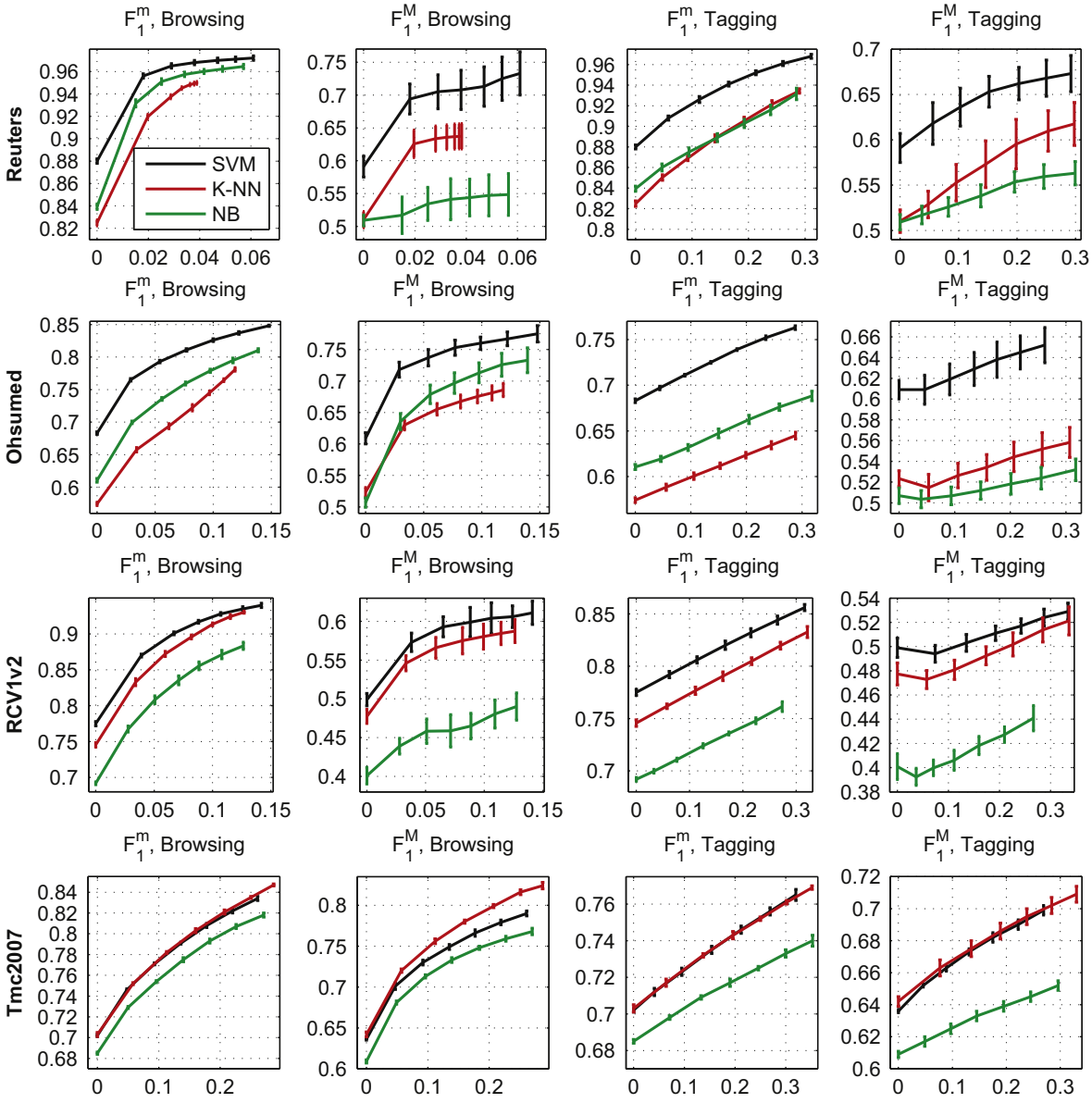

Fig. 1. Accuracy–rejection curves attained on the four data sets related to text categorisation (one row for each data set). Each plot shows the curves of the three base classifiers (SVM, k-NN, NB), for the four combinations of the considered cost models (browsing and tagging) and accuracy measures (micro and macro F). Average and standard deviation of the testing set accuracy is reported, over the ten runs of the experiments.

of the manual annotation cost of the whole Reuters data set. Similar relative increases of $\hat{F}_1^{\mathrm{m}}$ can be observed for the k-NN and NB classifiers (which are less accurate than SVMs), and on the other data sets, although at the expense of higher rejection rates (i.e., higher manual annotation cost).

For some data sets, the rejection rate did not reach the maximum allowed value, $r_{\max}=0.3$. In some cases (e.g., Reuters), this was due to the fact that no further increase of $\hat{F}_1^{\mathrm{m}}$ was attained by increasing r beyond a certain value. In other cases, this was due instead to the constraint $r_{\max,k}=r_{\max}$, and to the fact that the number of rejected decisions turned out to be skewed across classes. In particular, few decisions turned out to be rejected on rarer classes. In this case, to attain a desired rejection rate $r_{\max}$, one should either choose a different $r_{\max,k}$ for different classes, taking into account class frequency, or choose values of $r_{\max,k}$ such that $\sum_{k=1}^{N} r_{\max,k} > C_{\max}/t_{\mathrm{d}}$, as explained in Section 4.2.

Similar results were attained for the $\hat{F}_1^{\mathrm{M}}$ measure (Figs. 1 and 2, second column). The main difference is that, in data sets containing many rare classes (see Table 2), $\hat{F}_1^{\mathrm{M}}$ exhibits a higher variance. The reason is that label-wise macro-averaged measures are dominated by the accuracy on rare classes, which exhibits a higher variance than the one of common classes. We also point out that the accuracy (especially the macro F) attained by the considered classifiers on Corel-5k is rather poor: in this case, the improvements attained by a reject option may be not sufficient.

We finally evaluated the computational cost of Algorithm 1, when $\hat{A}=\hat{F}_1^{\mathrm{m}}$. According to online Appendix D, Algorithm 1 executes up to $N(M+1)$ iterations of its repeat-until loop, and evaluates $\hat{F}_1^{\mathrm{m}}$ for up to $N^2(M+1)^2(M+2)/2$ different values of the $2N$ thresholds. Consider now the Ohsumed data set, which contains the highest number of classes N and of training samples M (see Table 2). Using the SVM classifier, on average 5.1 iterations of the repeat-until loop of Algorithm 1 were carried out, while $\hat{F}_1^{\mathrm{m}}$ was evaluated on average for $0.7N(M+1)(M+2)$ times: both values are much lower than the corresponding upper bounds. Similar results were obtained for the other classifiers and data sets. This provides evidence that Algorithm 1 scales very well with respect to the number of classes and of samples.

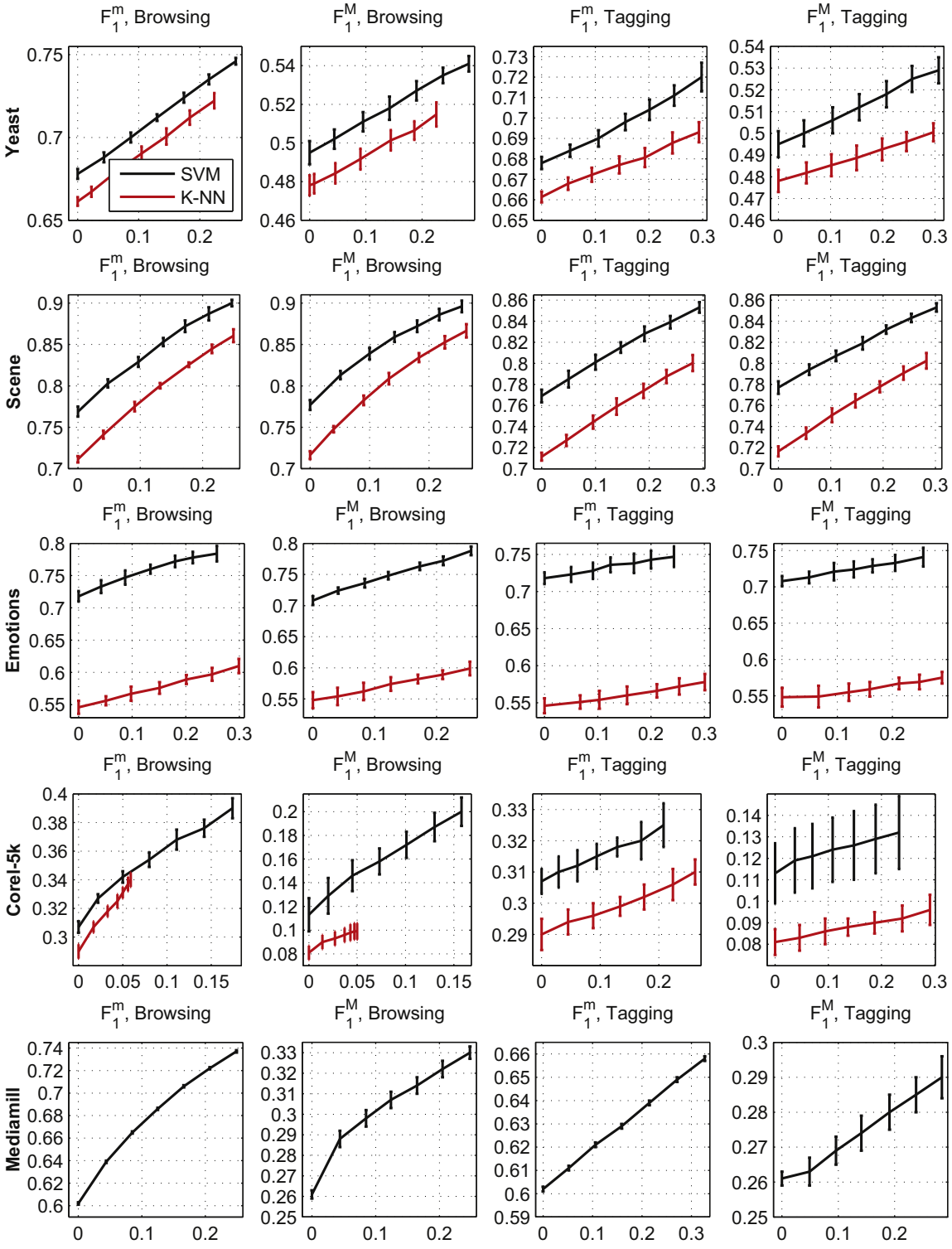

Fig. 2. Accuracy–rejection curves attained on the five data sets not related to text categorisation (one row for each data set). See caption of Fig. 1 for the details.

5.3. Results for the approximated tagging cost model

In these experiments, we computed the reliability measures $R(\mathbf{x})$ of Eqs. (20) and (21) by estimating the corresponding terms, respectively $TP(\{\mathbf{x}\})$, $FP(\{\mathbf{x}\})$, $FN(\{\mathbf{x}\})$, and $TP_k(\{\mathbf{x}\})$, $FP_k(\{\mathbf{x}\})$, $FN_k(\{\mathbf{x}\})$, from the score distributions evaluated on training samples. To this aim, for each class $k \in \{1,\ldots,N\}$ we computed 20-bins histograms of the corresponding TP, FP and FN distributions as functions of the scores $s_k(\cdot)$. For instance, a testing sample $\mathbf{x}$ for which $f_k(\mathbf{x}) = +1$, can be either a TP or a FP for class k. Accordingly, we set $FN_k(\{\mathbf{x}\}) = 0$, and computed $TP_k(\{\mathbf{x}\})$ as $\mathbb{P}[Y_k = +1 | s_k(\mathbf{x}), f_k(\mathbf{x}) = +1]$, which was estimated as the bin value including $s_k(\mathbf{x})$ in the TP histogram of class k, and similarly for $FP_k(\{\mathbf{x}\})$.

Figs. 1 and 2 (third and fourth columns) show the accuracy–rejection curves attained by using the implementation of a reject option of Section 4.3. Recall that, under cost model (11), manual annotation cost is proportional to the fraction r of rejected

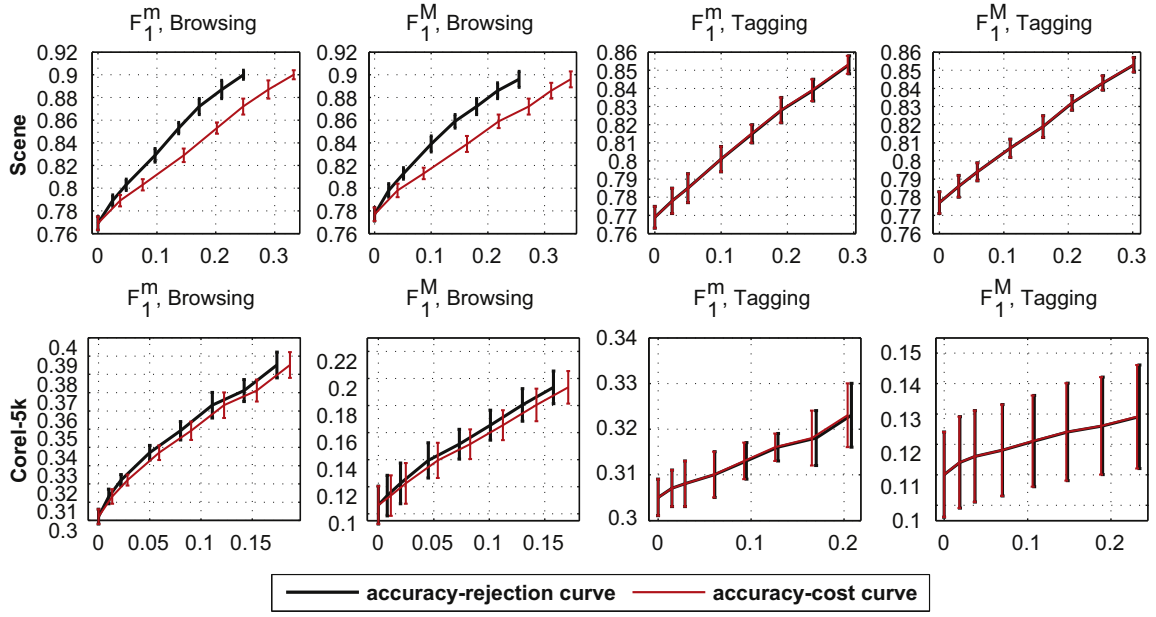

Fig. 3. Comparison between accuracy–rejection and accuracy–cost curves on Scene and Corel-5k, attained using a SVM as base classifier, for all combination of accuracy measures (either micro or macro F) and image annotation technique (either tagging or browsing). Average and standard deviation of testing set accuracy is reported, over the ten runs of the experiments. The accuracy–rejection curves are the same as in Figs. 1 and 2.

samples (rejection rate). As in Section 5.2, we considered values of r_{max} in [0.0,0.3], with steps of 0.05.

It can be seen that $\hat{F}_1^m$ always increased for increasing rejection rates, although the relative improvement with respect to classifiers without a reject option (i.e., $r=0$) was lower than in case of rejection of decisions, being equal the rejection rate. Lower improvements can also be observed for $\hat{F}_1^M$, as well as a higher variance for some data sets, as in Section 5.2. Furthermore, for Ohsumed and RCV1v2, $\hat{F}_1^M$ did not always increase for increasing rejection rates. This may be due to the suboptimal reliability measure $R(\mathbf{x})$ defined in Section 4.3. Defining a more effective reliability measure remains thus an interesting open issue.

5.4. Comparison between the accuracy–cost and accuracy–rejection curves

We have seen that, in order to simplify learning problem (9), it could be necessary to approximate the underlying cost model. In particular, if the cost function (10) or (11), used respectively in learning problems (15) and (19), is an approximation of the actual cost function, then the attained rejection rate is not proportional to the corresponding manual annotation cost of rejections. In this case, the accuracy–rejection curve is only an approximation of the actual accuracy–cost curve. In this section we experimentally evaluate how much the two curves can differ. To this aim, one should know the exact cost model for the task at hand. However, in our experiments the real cost models (Eq. (6)) are known only for data sets related to image annotation, i.e., Scene and Corel-5k. We will thus consider only these data sets in the following. The values of the parameters of cost models (6), that have been empirically estimated in [1] using a real annotation tool, and expert users, are the following[3]: $t_s=5.6$ s. and $t_f=6.8$ s. for tagging; $t_p=1.4$ s. and $t_n=0.2$ s. for browsing. Fig. 3 shows the comparison between the accuracy–rejection curves attained on Scene and Corel-5k (the same as in Figs. 1 and 2), and the corresponding accuracy–cost curves. To ease the comparison, manual annotation cost is reported as the fraction of the cost needed for manually annotating the whole data set, and is thus shown in the same scale as the rejection rate.

Recall that the approximated browsing and tagging cost models of (10) and (11) have been derived under the assumptions that $t_p \approx t_n$, and $t_s \gg N_p(\mathbf{x})t_f$, respectively. Given the above values of the four parameters, and the average number of labels per sample $N_p(\mathbf{x})$ (1.06 ± 0.25 for Scene and 3.52 ± 0.66 for Corel, see Table 2), the above assumptions turn out to be violated by a rather large extent. Our approximations of the two cost models is thus rather inaccurate. However, the actual accuracy–cost curves of Fig. 3 are very close to the corresponding accuracy–rejection curves, in the case of tagging, and rather close in the case of browsing (the largest difference between the accuracy values in the two curves is below 0.04). These results suggest that even inaccurate approximations of the cost models may provide good estimates of the accuracy–cost curve.

6. Discussion and contributions

In this work we provided two main contributions: (i) We developed a framework for classification with a reject option in ML problems. We defined in particular how to evaluate the trade-off between accuracy and manual annotation cost, and the general form of the classifier learning problem. (ii) We developed two possible implementations of our framework for the widely used micro and (label-wise) macro F measure, and the only two cost models formalised so far for annotation tasks. Any ML classifier that outputs a real-valued score for each class can be used in our implementations.

This paper extends our previous works on the same topic [10–12], in the following respects. In [10] we extended the precision and recall measures to take into account the presence of rejected decisions, and proposed the strategy of rejection of decisions. In [10,12] we proposed the strategy of rejection of samples. However, in these works we did not formalise the goal of a reject option in terms of attaining a trade-off between classifier accuracy and manual annotation cost. We focused on the trade-off between accuracy and rejection rate instead,

[3] The values of these parameters can be affected by several factors, like the specific tool used [1]. For the purpose of our experiments, we can consider the values of [1].

without considering the underlying cost model. Moreover, only a suboptimal algorithm was devised in [10] to solve the corresponding learning problem, when the micro F accuracy measure is used, and a limited experimental evaluation was carried out. In [11] we considered a two-stage ML classifier architecture, in which only the first-stage classifier is allowed to withhold decisions. Rejected decisions are then handled by a more accurate and more costly second-stage classifier. In this context, our goal was to improve the trade-off between accuracy and processing time, while no manual annotation of rejections was involved.

Let us now discuss the issue of label dependence/correlation in ML problems, which has been addressed recently by several authors (a thorough analysis of this issue can be found in [27]). Taking into account label dependence (if any) can indeed improve the accuracy of ML classifiers. Our framework does not set any limitation on the possibility of taking label correlation into account: to this aim, a suitable choice of (i) the decision function and (ii) the cost function has to be made, in learning problem (9).

(i) With regard to the decision function, if it is implemented using a two-stage approach, in which a rejection criterion is defined on the outputs of any trained ML classifier (which is the strategy adopted in this paper, as in most SL problems), then a straightforward solution exists: one can use at the first stage any ML classifier (without a reject option) that takes correlation into account. We also point out that in our implementations of Section 4, the parameters of decision functions with a reject option (either the thresholds $t_1^{\mathrm{L}},t_1^{\mathrm{H}},\ldots,t_N^{\mathrm{L}},t_N^{\mathrm{H}}$, or the reliability measure $R(\cdot)$ and the corresponding threshold t) are computed not independently for each class, but taking into account all classes simultaneously, by directly maximising the considered performance measure. This allows us to (implicitly) take into account label correlation.

(ii) Label correlation (as well as other factors, like class frequency) may affect also the cost model, as pointed out in Section 2.2. Note however that the cost models of [1], that were used in this paper, do not take into account label correlation, nor other factors like class frequency, although they refer to real image annotation techniques, and have been validated in a realistic setting in [1]. Anyway, if these factors are taken into account in the definition of the cost function, they could lead to the same issue discussed in Section 3.3, namely, a learning problem for which it is difficult to develop an optimisation algorithm. In this case, the same solution proposed in Section 3.3 can be adopted, i.e., choosing a suitable decision function, and (if necessary), approximating beforehand the cost function.

We finally mention two future research directions. First, it is obviously interesting to develop implementations of our framework for other ML accuracy measures, besides micro and (label-wise) macro F, and for different cost models (if any) from the ones of image annotation with tagging and browsing. Clearly, this requires the definition of formal cost models for annotation tasks different from image annotation, which was out of the scope of this work. Second, the general form of the learning problem (9) for ML classifiers with a reject option was defined by considering the empirical estimate of classification accuracy as the objective function. It would be interesting to devise a proper regularisation term, to deal with overfitting in the case of small training set size.

Conflict of interest statement

None declared.

Acknowledgements

This work has been partly supported by a 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. Supplementary data

Supplementary data " associated with this paper can be found in the online version at http://dx.doi.org.10.1016/j.patcog.2013.01.035.

References

[1] R. Yan, A. Natsev, M. Campbell, An efficient manual image annotation approach based on tagging and browsing, in: ACM Multimedia Workshop on the Many Faces of Multimedia Semantics, 2007, pp. 13–20.
[2] A. Aronson, W. Rogers, F. Lang, A. Névéol, 2008 report to the board of scientific counselors, 〈http://ii.nlm.nih.gov〉, 2008.
[3] F. Sebastiani, Machine learning in automated text categorization, ACM Computing Surveys 34 (1) (2002) 1–47.
[4] L. Reeve, H. Han, Survey of semantic annotation platforms, in: Proceedings of the ACM Symposium on Applied computing, 2005, pp. 1634–1638.
[5] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, in: Data Mining and Knowledge Discovery Handbook, 2010, pp. 667–685.
[6] S. Nowak, M. Huiskes, New strategies for image annotation: overview of the photo annotation task at ImageCLEF 2010, Working Notes of CLEF 2010.
[7] M. Ruiz, A. Aronson, User-centered evaluation of the Medical Text Indexing (MTI) system, 〈http://ii.nlm.nih.gov〉, 2007.
[8] C.K. Chow, An optimum character recognition system using decision function, IEEE Transactions on Computers 6 (4) (1957) 247–254.
[9] C.K. Chow, On optimum recognition error and reject tradeoff, IEEE Transactions on Information Theory 16 (1) (1970) 41–46.
[10] G. Fumera, I. Pillai, F. Roli, Classification with reject option in text categorisation systems, in: Proceedings of the International Conference on Image Analysis and Processing, 2003, pp. 582–587.
[11] G. Fumera, I. Pillai, F. Roli, A two-stage classifier with reject option for text categorisation, in: Proceedings of the International Workshop on Statistical Techniques in Pattern Recognition, Lecture Notes in Computer Science, vol. 3138, Springer, 2004, pp. 771–779.
[12] I. Pillai, G. Fumera, F. Roli, A classification approach with a reject option for multi-label problems, in: Proceedings of the International Conference on Image Analysis and Processing, Lecture Notes in Computer Science, vol. 6978, Springer, 2011, pp. 98–107.
[13] E. Grall, P. Beauseroy, A. Bounsiar, Quality assessment of a supervised multilabel classification rule with performance constraints, in: Proceedings of the 14th European Signal Processing Conference (EUSIPCO 2006), 2006.
[14] T. Pietraszek, On the use of ROC analysis for the optimization of abstaining classifiers, Machine Learning 68 (2007) 137–169.
[15] F. Tortorella, A ROC-based reject rule for dichotomizers, Pattern Recognition Letters 26 (2005) 167–180.
[16] Y. Yang, A study of thresholding strategies for text categorization, in: Proceedings of the International Conference on Research and Development in Information Retrieval, 2001.
[17] R.-E. Fan, C.-J. Lin, A Study on Threshold Selection for Multi-Label, Technical Report, National Taiwan University, 2007.
[18] I. Pillai, G. Fumera, F. Roli, Threshold optimisation for multi-label classifiers, Pattern Recognition, http://dx.doi.org/10.1016/j.patcog.2013.01.012, in press.
[19] J.R. Quevedo, O. Luaces, A. Bahamonde, Multilabel classifiers with a probabilistic thresholding strategy, Pattern Recognition 45 (2) (2012) 876–883.
[20] D.D. Lewis, et al., RCV1: a new benchmark collection for text categorization research, Journal of Machine Learning Research 5 (2004) 361–397.
[21] W.R. Hersh, C. Buckley, T.J. Leone, D.H. Hickam, Ohsumed: an interactive retrieval evaluation and new large test collection for research, in: Proceedings of the ACM SIGIR Conference, ACM/Springer, 1994, pp. 192–201.
[22] G. Tsoumakas, et al., Mulan: a java library for multi-label learning, Journal of Machine Learning Research 12 (2011) 2411–2414.
[23] J. Read, et al., Multi-label classification using ensembles of pruned sets, in: Proceedings of the International Conference on Data Mining, 2008, pp. 995–1000.
[24] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 〈http://www.csie.ntu.edu.tw/∼cjlin/libsvm〉, 2001.
[25] Y. Yang, X. Liu, A re-examination of text categorization methods, in: Proceedings of the 22nd Annual International ACM/SIGIR Conference Research and Development in Information Retrieval, 1999, pp. 42–49.
[26] J.D. Rennie, et al., Tackling the poor assumptions of Naive Bayes text classifiers, in: Proceedings of the International Conference on Machine Learning, 2003, pp. 616–623.
[27] K. Dembczynski, W. Waegeman, W. Cheng, E. Hüllermeier, On label dependence and loss minimization in multi-label classification, Machine Learning 88 (2012) 5–45.

Ignazio Pillai received the M.Sc. degree in Electronic Engineering, with honours, and the Ph.D. degree in Electronic Engineering and Computer Science from the University of Cagliari, Italy, in 2002 and 2007, respectively. Since 2003 he has been working for the Department of Electrical and Electronic Engineering at the University of Cagliari, Italy, where he is now a post doc in the research group on Pattern Recognition and Applications led by Prof. Fabio Roli. From 2007 to 2009 he has been a junior fellow at the Ambient Intelligence Laboratory of the Science and Technology Park, in Sardinia, Italy, where he developed algorithms and tools for spam filtering, face verification and image categorisation. His research interests are related to methodologies and applications of statistical pattern recognition. His main research topics are currently multi-label classification, multimedia document categorisation and classification with a reject option. He serves as a reviewer for several international conferences and journals, including Pattern Recognition Letters and Pattern Analysis and Applications. He is a member of the Italian chapter of the International Association for Pattern Recognition.

Giorgio Fumera received the M.Sc. degree in Electronic Engineering, with honors, in 1997, and the Ph.D. degree in Computer Engineering, in 2002, from the University of Cagliari, Italy. Since 2011 he is Associate Professor on Computer Engineering at the University of Cagliari, Italy. He is a member of the Pattern Recognition and Application research group at the Department of Electrical and Electronic Engineering, and cooperates with the Ambient Intelligence Laboratory of the Science and Technology Park, in Sardinia, Italy. His research activity is currently focused on multiple classifier systems, classification with a reject option, and theory and design methods of pattern recognition systems in adversarial environments, with applications to text and image categorisation, spam filtering and person re-identification techniques for video surveillance tasks. On these topics he published ten papers on international journals, two book chapters and more than forty papers on international conferences and workshops. He is a member of the Institute of Electrical and Electronics Engineers (IEEE), the IEEE Computer Society, the Italian chapter of the International Association for Pattern Recognition (IAPR), and the Italian Association for Artificial Intelligence (AI*IA). He is associate editor of the international journal "Electronic Letters on Computer Vision and Image Analysis", and acts as a reviewer for the main international journals and conference of the statistical pattern recognition field.

Fabio Roli received his M.S. degree, with honours, and Ph.D. degree in Electronic Engineering from the University of Genoa, Italy. He was a member of the research group on Image Processing and Understanding of the University of Genoa, Italy, from 1988 to 1994. He was adjunct professor at the University of Trento, Italy, in 1993 and 1994. In 1995, he joined the Department of Electrical and Electronic Engineering of the University of Cagliari, Italy, where he is now professor of computer engineering and head of the research group on pattern recognition and applications. His current research activity is focused on multiple classifier systems and their applications to biometric personal identification, multimedia text categorisation, and computer security. On these topics, he has published more than one hundred papers at conferences and on journals. He was a very active organiser of international conferences and workshops, and established the popular workshop series on multiple classifier systems. He serves as member in many panels of funding agencies, including the NATO advisory panel on Information and Communications Security. He is a member of the governing boards of the International Association for Pattern Recognition and of the IEEE Systems, Man and Cybernetics Society. He is Fellow of the IEEE, and Fellow of the International Association for Pattern Recognition.