Learning of multilabel classifiers

Ignazio Pillai, Giorgio Fumera, Fabio Roli
Dept. of Electrical and Electronic Engineering, University of Cagliari
Piazza d'Armi, 09123 Cagliari, Italy
Email: {pillai,fumera,roli}@diee.unica.it Web: http://pralab.diee.unica.it

***Abstract*—Developing learning algorithms for multilabel classification problems, when the goal is to maximizing the micro-averaged F measure, is a difficult problem for which no solution was known so far. In this paper we provide an exact solution for the case when the popular binary relevance approach is used for designing a multilabel classifier. We prove that the empirical maximum of the micro-averaged F measure can be attained by iteratively retraining class-related binary classifiers whose learning algorithm is capable of maximizing a modified version of the F measure of a two-class problem. We apply our optimization strategy to an existing formulation of support vector machine classifiers tailored to performance measures like F, and evaluate it on benchmark multilabel data sets.**

I. Introduction

In binary classification tasks characterized by a high class imbalance, or related to information retrieval, performance measures based on some trade-off between precision and recall are more suitable than accuracy. A common choice is the F measure, which is defined as a weighted harmonic mean of precision and recall [1]. Developing learning algorithms that maximize the F measure is however difficult, as it does not decompose over samples, contrary to measures like classification accuracy. Only a few such learning algorithms have been proposed so far: variants of the support vector machine (SVM) [2] and logistic regression classifier [3], whose objective function is however non-convex, and an extension of the standard SVM formulation, that maximizes a convex lower bound of a broad class of performance measures, including F [4]. A different approach, first proposed in [5], consists of assigning to a *given* set of *testing* samples the labels that maximize the expected value of the F measure, with respect to the distribution $P(\mathbf{x}, y)$, where $\mathbf{x}$ denotes a feature vector and $y \in \{-1, +1\}$ its class label. In this inference approach, training samples are used for estimating the distribution $P(\mathbf{x}, y)$. Exact inference algorithms, with different computational complexities, have been developed by several authors (see [6], [7] and references therein).

The above F measure is defined for binary problems, and can be named *class-wise*. For multilabel problems, where each sample can belong to more than one class, three different extensions have been defined: *sample-wise*, which decomposes over samples and is computed like the class-wise F, but over the labels of a single sample; *macro-averaged* (denoted in the following as F^{M}), which is the mean of the class-wise F of each class; and *micro-averaged* (F^{m}), which is computed after pooling per-sample labels across categories. The sample-wise F decomposes over samples, and the macro-averaged F decomposes over labels. Learning algorithms that aim at maximizing them have been proposed respectively in [8], [9], and in [10]. Inference algorithms that maximize the expected value of the sample-wise F have also been derived in [11] and [12]. The micro-averaged F does not decomposes over samples nor over labels, instead, and is thus the most difficult measure to optimize. Neither learning nor inference algorithms have been developed yet for this measure.

In this paper we develop the first learning algorithm known so far, that is capable to maximize the F^{m} measure. Our algorithm applies to the case when the popular *binary relevance* approach is used for designing a multilabel classifier, i.e., a distinct binary classifier is used for each class, and the learning algorithm of binary classifiers is in turn capable of maximizing an ad hoc variant of the class-wise F measure. Building on a property of F^{m} derived in our previous work [13], summarized in Sect. II, in Sect. III we show that the global maximum of F^{m} can be attained by iteratively retraining each binary classifier. In Sect. IV we show that one instance of the required kind of binary learning algorithm can be derived from the SVM formulation of [4]. In Sect. V we empirically evaluate on benchmark data sets the multilabel classifier obtained by applying our learning algorithm to this SVM formulation. Interesting directions for future work are discussed in Sect. VI.

II. Background and previous work

We denote the d-dimensional feature space of a two-class problem as $X \subseteq \mathbb{R}^d$, the class labels as $Y = \{-1, +1\}$, a feature vector as $\mathbf{x} \in X$, and the corresponding label as $y \in Y$. Precision (p) and recall (r) of a binary classifier $f : X \mapsto Y$ can be empirically estimated on a data set of n samples from the counts of true positives (TP), false positives (FP) and true negatives (TN), as $p = TP/(TP + FP)$ and $r = TP/(TP + FN)$. The class-wise F is defined as a weighted harmonic mean of p and r:

$$F_\beta = \frac{1 + \beta^2}{\frac{1}{p} + \frac{\beta^2}{r}} = \frac{(1 + \beta^2)TP}{(1 + \beta^2)TP + FP + \beta^2 FN} , \quad (1)$$

where $\beta \in [0, +\infty)$. In a multilabel problem each sample can belong to one or more of the N classes. The micro-averaged F measure is defined as:

$$F_\beta^{\mathrm{m}} = \frac{(1 + \beta^2) \sum_{k=1}^N TP_k}{\sum_{k=1}^N [(1 + \beta^2)TP_k + FP_k + \beta^2 FN_k]} . \quad (2)$$

The simplest approach for designing a multilabel classifier $f : X \mapsto Y^N$ is binary relevance [14]: it consists of independently training N binary classifiers $f_k : X \mapsto Y$, that predict if an input sample belongs or not to the corresponding class. The decision functions f_k are often obtained by thresholding a real-valued discriminant function g_k: $f_k(\mathbf{x}) = \mathrm{sign}(g_k(\mathbf{x}))$. Since most learning algorithms maximize classification accuracy, a common approach for improving the performance of a multilabel classifier is tuning its decision thresholds θ_k to maximize the desired measure, on a data set S (e.g., a validation set, or by cross-validation on training samples), such that $f_k(\mathbf{x}) = \mathrm{sign}(g_k(\mathbf{x}) - \theta_k)$ [15], [16]. This optimization problem can be solved efficiently for F_β^{M}, since it decomposes over classes. This is not the case of F_β^{m}, instead: in principle, only an exhaustive search can provide the optimal threshold values, with an infeasible computational complexity of $O(n^N)$ (where n denotes the size of S) [13]. Until [13], only two suboptimal solutions had been proposed: using the same threshold values that maximize F_β^{M} [15], and a heuristic iterative optimization algorithm [16].

In [13] we developed an optimization strategy that finds the optimal threshold values in $O(n^2N^2)$ time, based on the derivation of the following property of F_β^{m} as a function of the decision thresholds $\theta = (\theta_1, \ldots, \theta_N)$, on a given data set S:

Property 1. *Consider a data set S and any given set of threshold values θ. If, for each $k = 1, \ldots, N$, $\arg\max_{\theta'_k} F_\beta^{\mathrm{m}}(\theta_1, \ldots, \theta_{k-1}, \theta'_k, \theta_{k+1}, \ldots, \theta_N) = \theta_k$, then the absolute maximum of F_β^{m} on S equals $F_\beta^{\mathrm{m}}(\theta)$.*

This implies that the optimal threshold values can be found by an iterative optimization strategy that increases F_β^{m} with respect to a *single* threshold at a time. More precisely, θ can be initially set to any value; then, a loop over all classes $k = 1, \ldots, N$ is carried out, and each θ_k is updated to *any* value θ_k^* that provides an increase of F_β^{m} (if any), keeping the other thresholds fixed at their current values; this is repeated until no increase can be attained in a whole loop over all classes. To speeding up convergence, one can choose at each step the θ_k^* value that maximizes F_β^{m}. The expression of F_β^{m} to be maximized at each step as a function of a single threshold θ_k is:

$$F_\beta^{\mathrm{m}}(\theta_k) = \frac{(1+\beta^2)TP_k + N_k}{(1+\beta^2)TP_k + FP_k + \beta^2 FN_k + D_k}\ , \qquad (3)$$

where the constants N_k and D_k denote the contribution to F_β^{m} of the other *current* classifiers (see Eq. 2):

$$N_k = (1+\beta^2)\sum_{j\neq k} TP_j, \qquad (4)$$

$$D_k = \sum_{j\neq k}[(1+\beta^2)TP_j + FP_j + \beta^2 FN_j]. \qquad (5)$$

III. Learning algorithms for F_β^{m}

Our analysis in [13] focused on the case when F_β^{m} is computed as a function of the decision thresholds of previously trained binary classifiers f_k. However, a fresh look to the proof of Property 1 (see [13]) reveals that it has a much broader scope, since it involves only the values of F_β^{m} as functions of the class labels assigned by the decision functions f_k, *regardless* of how these are implemented. Property 1 is thus valid also in the particular case when θ_k denotes the parameters of f_k, that are set by its learning algorithm $\mathcal{L}$ (e.g., the coefficients $\theta_k = (\mathbf{w}_k, b_k)$ of a linear classifier $f_k(\mathbf{x}) = \mathrm{sign}(\mathbf{w}_k^t\mathbf{x} + b_k)$). Also in this case the above optimization strategy provides the global maximum of F_β^{m}, provided that $\mathcal{L}$ is capable of maximizing (3) as its objective function. This corresponds to finding the θ_k^* value that maximizes F_β^{m}, keeping the other θ_j's fixed, as described in Sect. II.

The above optimization procedure is formally described by Algorithm 1, where S_k denotes the data set obtained from S by setting to $+1(-1)$ the label of samples that (do not) belong to class k; and $\mathcal{L}(S_k; N_k, D_k)$ denotes the application of $\mathcal{L}$ to S_k, to learn the classifier f_k, by maximizing the objective function (3) with given values of N_k and D_k. According to Sect. II, any initial value can be assigned to each θ_k; a reasonable choice is to set them by maximizing the corresponding class-wise F measure (1), i.e., $\theta_k \leftarrow \mathcal{L}(S_k, 0, 0)$, which amounts to maximizing F_β^{M}. Note that also Algorithm 1 is guaranteed to converge in a finite number of steps, since each f_k is updated only if this provides an increase of F_β^{m}, which can happen only for a finite number of times on a finite data set.

Our theoretical result provides an *exact* solution to the problem of developing multilabel learning algorithms using the binary relevance approach, capable to maximize F_β^{m} on training samples. It shows that, although F_β^{m} does not decompose over classes, its global maximum can be attained by an iterative optimization strategy that does exploit a decomposition over classes, provided that a *binary* learning algorithm capable of maximizing Eq. (3) is available. In other words, this reduces the problem of maximizing the multilabel F_β^{m} measure to the problem of maximizing a two-class performance measure similar to the class-wise F. We now discuss some practical issues.

Eq. (3) can be maximized as a function of a decision threshold in $O(n)$ time [13]. Maximizing it as a function of all the parameters of a given classifier (the step $\mathcal{L}(S_k; N_k, D_k)$ of Algorithm 1) could be infeasible instead, since it is a discrete measure computed from error counts. Nevertheless, our optimization strategy converges to the global maximum of F_β^{m}, even if a higher F_β^{m} value (if any) is found at each step with respect to θ_k, not necessarily the highest one (see Sect. II). However, this is not guaranteed either, if $\mathcal{L}$ maximizes an approximation of the target measure, which is inevitable for computational reasons in the case of discrete measures. The consequence is that, in practice, our optimization strategy does not guarantee to provide the global maximum of F_β^{m}, similarly to all learning algorithms tailored to classification accuracy. The convergence of Algorithm 1 in a finite number of steps is nevertheless guaranteed, for the same reasons explained above.

To exploit our result, binary learning algorithms capable of maximizing (a suitable approximation of) the objective function (3) are required. As mentioned in Sect. II, two only learning algorithms that maximize a measure similar to (3), i.e., (1), exist [3], [4]. Note that (1) in itself is *not* a suitable approximation of (3) as it may seem, since our optimization strategy requires that the objective function (3) is tuned at each step to the *current* value of F_β^{m} through the parameters N_k and D_k. Nevertheless, it turns out that the objective function used in [3], [4] can be easily modified into (3). For lack of space, in the following sections we will focus only on [4].

The computational complexity of our optimization strategy depends on the kind of binary classifier and on the corresponding learning algorithm $\mathcal{L}$. We leave a theoretical analysis to future work, and evaluate the processing cost empirically in Sect. V. Here we point out that Algorithm 1 requires to run $\mathcal{L}$

Algorithm 1 Maximization of F_β^{m}

```
Input: a $N$-class multilabel training set $S$; a learning algorithm $\mathcal{L}$ for binary classifiers
Output: binary classifiers $f_k(\mathbf{x};\theta_k)$, $k = 1,\ldots,N$
  for $k = 1,\ldots,N$ do
    $\theta_k \leftarrow \mathcal{L}(S_k; 0, 0)$
  end for
  repeat
    $updated \leftarrow$ False
    for $k = 1,\ldots,N$ do
      Compute $N_k$ and $D_k$ using Eqs. (4) and (5)
      $\theta_k^* \leftarrow \mathcal{L}(S_k; N_k, D_k)$
      if $F_\beta^{\mathrm{m}}(\theta_1,\ldots,\theta_{k-1},\theta_k^*,\theta_{k+1},\ldots,\theta_N) > F_\beta^{\mathrm{m}}(\theta)$ then
        $\theta_k \leftarrow \theta_k^*,\ updated \leftarrow$ True
      end if
    end for
  until $updated =$ False
  return $f_k(\mathbf{x};\theta_k)$, $k = 1,\ldots,N$
```

at least for $2N$ times (i.e., two complete loops over all classes), while the standard binary relevance approach requires only N runs. Therefore, an additional stopping condition should be used in practice, e.g., setting a threshold on the maximum number of **repeat** loops, or on the relative increase of F_β^{m} in subsequent loops. Nevertheless, empirical evidences in Sect. V suggest that two complete loops over all classes are sufficient to approach the global maximum of same F_β^{m}. The processing cost could be further reduced by using, since the second loop over classes, incremental learning techniques.

The processing cost can depend also on the order in which the classes are scanned in the **for** loop, as shown (for the case when only the decision thresholds are updated) in [13]. Also the resulting multilabel classifier can depend on the class ordering, in the case when $\mathcal{L}$ does not guarantee to find the global maximum of the objective function (3), or an additional stopping condition is used. Devising suitable heuristics for choosing an effective class ordering is left to future work.

IV. Application to Support Vector Machines

Here we show how the SVM formulation of [4], named SVM$^{\Delta}_{\text{multi}}$, can be exploited in our framework.

SVM$^{\Delta}_{\text{multi}}$ is an extension of the standard SVM learning algorithm to a class of performance measures that not decompose into expectations over samples, including the class-wise F measure (1). While standard SVMs learn a decision rule that predicts the label of a single sample, the learning problem of SVM$^{\Delta}_{\text{multi}}$ was formulated as a multivariate prediction of all the n samples in the training set. Denoting as $\overline{\mathbf{x}} \in X^n$ and $\overline{y} \in Y^n$ the tuples $(\mathbf{x}_1,\ldots,\mathbf{x}_n)$, and $(y_1,\ldots,y_n)$, a multivariate decision rule $\overline{h} : X^n \mapsto Y^n$ was defined as: $\overline{h}(\overline{\mathbf{x}}) = \arg\max_{\overline{y}' \in Y^n}\{\mathbf{w}^t\Psi(\overline{\mathbf{x}},\overline{y}')\}$, where Ψ is intended as a function that returns a matching score vector between $\overline{\mathbf{x}}$ and $\overline{y}'$. In [4] it was defined as: $\Psi(\overline{\mathbf{x}},\overline{y}') = \sum_{i=1}^{n} \overline{y}'\mathbf{x}_i$. The learning problem was formulated accordingly as the convex optimization problem (6), where $\overline{\mathbf{x}}$ and $\overline{y}$ refer to training samples, and $\Delta(\overline{y}',\overline{y})$ denotes the loss function, that in our case can be defined as $(1 - F) \in [0, 1]$:

$$\begin{array}{ll} \min_{\mathbf{w},\xi\geq 0} & \frac{1}{2}\|\mathbf{w}\|^2 + C\xi \\ \text{s.t.} & \forall \overline{y}' \in Y^n \setminus \overline{y} : \\ & \mathbf{w}^t[\Psi(\overline{\mathbf{x}},\overline{y}) - \Psi(\overline{\mathbf{x}},\overline{y}')] \geq \Delta(\overline{y}',\overline{y}) - \xi \end{array} \qquad (6)$$

Even if the optimization problem (6) has $2^n - 1$ constraints, it can be solved efficiently when Δ can be computed in polynomial time from TP, FP and FN counts, which is the case of the F measure (1). In this case enumerating all $2^n - 1$ tuples $\overline{y}' \in Y^n \setminus \overline{y}$ is not necessary; it suffices instead enumerating all distinct values that $\Delta(\overline{y}',\overline{y})$ can attain on training samples, which are only $O(n^2)$ [4]. Note that SVM$^{\Delta}_{\text{multi}}$ is a generalization of standard SVMs, since their learning problems coincide when $\Delta(\overline{y}',\overline{y})$ equals the number of training set errors $FP + FN$ [4].

Modification of the SVM$^{\Delta}_{\text{multi}}$ objective function. Similarly to the F measure (1), it is easy to see that also the measure (3) can be computed in polynomial time from TP, FP and FN counts of class k, for fixed values of N_k and D_k. Indeed, it differs from (1) only in the two additive constants N_k and D_k at the numerator and denominator, and thus it takes on $O(n^2)$ distinct values as well as (1). Therefore, it belongs to the category of performance measures to which the SVM$^{\Delta}_{\text{multi}}$ formulation applies. The same learning algorithm of SVM$^{\Delta}_{\text{multi}}$ can thus be used to maximize the objective function (3) in Algorithm (1).

Note that, similarly to standard SVMs, the learning problem (6) minimizes a convex upper bound of the loss function, and is thus not guaranteed to provide its global minimum on the training set. Therefore, when Algorithm 1 is applied to the modified SVM$^{\Delta}_{\text{multi}}$ formulation, it could converge to a non-global maximum of F_β^{m} on training samples, as explained in Sect. III.

V. Experiments

In this section we evaluate our learning algorithm on seven benchmark multilabel data sets. We use the binary classifier SVM$^{\Delta}_{\text{multi}}$, modifying its implementation[1] such that its objective function becomes Eq. 3. To this aim, we added the values of N_k and D_k to the command line parameters, and replaced the performance measure (1) with (3).[2] Note that the modified learning algorithm coincides with the original one, when $N_k = D_k = 0$. We implemented Algorithm 1 without additional stopping conditions.

Experimental set-up. We used the following data sets, related to four different domains (see Table I): Reuters 21578, the Heart Disease sub-tree of Ohsumed, the five subsets of Reuters RCV1v2, and the SIAM Text Mining Competition 2007 data set (text categorization); Scene (image annotation); Yeast (gene annotation); Emotions (music annotation).[3] For Reuters and Ohsumed we used tf–idf features, and carried out stemming, stop-word removal, and a further feature selection step using the information gain criterion. For the other data sets we used the feature vectors available at the mentioned URLs.

[1] http://www.cs.cornell.edu/People/tj/svm_light/svm_perf.html

[2] The implementation of SVM$^{\Delta}_{\text{multi}}$ also requires that the performance measure equals 1 when $FN_k = FP_k = 0$, and $TP_k \neq 0$. This is not true for (3), if $N_k \neq D_k$. To meet this requirement we divided (3) by the maximum value it can attain on training samples, corresponding to $TP_k = n_k^+, FP_k = FN_k = 0$, where n_k^+ denotes the number of positive samples for class k.

[3] Reuters: http://www.daviddlewis.com/resources/testcollections/reuters21578/; Ohsumed: http://trec.nist.gov/data/t9_filtering.html; Emotions: http://mulan.sourceforge.net/datasets.html; the other data sets are available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html.

Dataset	Samples (training/testing)	Features	Classes	Class frequency (min/max)
Reuters	7769 / 3019	18157	90	1E-4/0.37
Ohsumed	12775 / 3750	17341	99	2E-4/0.25
RCV1v2	3000 / 3000	47237	101	3E-4/0.46
TMC 2007	21519 / 7077	30438	22	0.01/0.60
Yeast	1500 / 917	104	14	0.06/0.75
Scene	1211 / 1196	295	6	0.14/0.23
Emotions	391 / 202	72	6	0.30/0.43

TABLE I. CHARACTERISTICS OF THE DATA SETS.

All data sets, except for Reuters RCV1v2, are subdivided into a training set and a testing set. We used as training samples a subset of 80% of the original training set, and used the remaining 20% as a validation set for parameter estimation; classification performance was evaluated on the original testing set. Ten runs were carried out, on random 80/20 subdivisions of the original training set. Reuters RCV1v2 is subdivided into five pairs of training and testing sets; we repeated our experiments on each subdivision, using a random 80/20 splitting of each original training set as above, and evaluated classification performance on the corresponding testing sets. For all data sets we used a linear kernel. The only hyper-parameter is C, which was chosen at each run among the values $\{10^{-1}, 10^0, 10^1, 10^2, 10^3\}$, separately for each class, by maximizing the corresponding F measure (1) on validation samples. Note that this parameter estimation procedure is tailored to F_β^M, and is thus suboptimal for our classifier. In the experiments we considered only the F_1^m measure ($\beta = 1$).

Classification performance. Since no other learning algorithm capable of maximizing F_1^m exists, we compared the performance attained by our algorithm with the baseline binary relevance approach, using binary classifiers trained with a standard learning algorithm, i.e., by minimizing the misclassification probability. To this aim, we used the standard SVM implementation available in the $\mathrm{SVM}^\Delta_\mathrm{multi}$ software. For the sake of completeness, we also evaluated the improvement of F_1^m attained by our algorithm with respect to the value obtained after a single loop over all classes, which amounts to maximizing F_1^M on training samples (i.e., to using the original $\mathrm{SVM}^\Delta_\mathrm{multi}$ classifier). The corresponding testing set F_1^m values are reported in Table II.

On the four text categorization data sets, optimizing F_1^m on training samples by our learning algorithm provided a slightly better testing set F_1^m value on Ohsumed, with respect to the standard binary relevance approach, while the opposite happened on TMC. The performances are instead comparable on Reuters and RCV1v2. Our algorithm outperformed binary relevance on the other three data sets. This provides evidence that, if F_1^m is the target performance measure, a learning algorithm that directly optimizes (a suitable approximation of) it, instead of the misclassification probability, can be advantageous.

Looking at the testing set F_1^m values attained by our algorithm after the first loop over all classes, it can be seen that they were improved during subsequent loops on five data sets, while no appreciable improvement was attained only on Scene and Emotions (where the average difference in F_1^m was lower than the standard deviation). This is reasonable, since these data sets do not exhibit a significant class imbalance (see Table I), and it is known that the difference between the micro- and macro-averaged F measure emerges especially when there are rare classes.

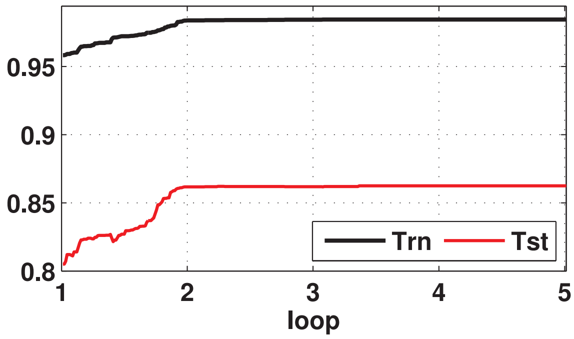

Fig. 1. F_1^m values attained on training and testing samples by Algorithm 1 in a single run on Reuters, during the loops over the 90 classes.

Processing cost. We evaluated the processing cost of Algorithm 1 in terms of the number of loops over all classes, i.e., on the number of runs of the underlying learning algorithm for binary classifiers. We remind the reader that the standard binary relevance approach (that does not maximize F_β^m) carries out only one loop. Table II (row "Loops") shows that Algorithm 1 converged after 4.0 to 7.7 loops on average, and that this value tends to be higher for data sets with a higher number of classes and of training samples. In some applications such a processing cost can be worth the improvement attainable on F_β^m, e.g., when there are rare classes (see above). Nevertheless, by analyzing the convergence speed we found that, already at the end of the first **repeat** loop, a very close F_1^m value was attained on training samples, as the one provided at convergence. A representative example is reported in Fig. 1, where the corresponding behavior of F_1^m on testing samples is also shown for completeness. This suggests that Algorithm 1 can be safely stopped after two only loops over all classes, with a processing time just twice as the one of binary relevance. For the sake of completeness, Table II shows also the overall number of binary classifiers updated in the inner **for** loop (row "Updates"). As one could expect, it tends to be higher for data sets in which a higher improvement over the baseline F_1^m value is attained. However, for the reason explained above, such updates mostly occurred during the first **repeat** loop (see also Fig. 1).

VI. CONCLUSIONS

We presented the first theoretical result derived so far about the development of multilabel learning algorithms capable of maximizing F_β^m. We showed that, when the binary relevance approach is used, the problem of maximizing F_β^m on a given data set reduces to the problem of maximizing a variant of the class-wise F measure of a two-class problem, despite F_β^m does not decompose over classes (nor over samples). We then devised the corresponding optimization algorithm. This provides a framework for developing learning algorithms tailored to F_β^m, since it can be applied to *any* binary learning algorithm capable of maximizing the mentioned variant of the class-wise F measure. Two such binary learning algorithms can be easily obtained from existing SVM and LR formulations tailored to the class-wise F. Our results could stimulate further efforts

	Data set						
	Reuters	Ohsumed	RCV1v2	TMC 2007	Yeast	Scene	Emotions
Optimized measure							
Error rate	0.857 ± 0.003	0.485 ± 0.007	0.643 ± 0.033	0.526 ± 0.005	0.610 ± 0.020	0.630 ± 0.023	0.649 ± 0.018
F_1^{M}	0.800 ± 0.005	0.443 ± 0.006	0.612 ± 0.017	0.468 ± 0.003	0.612 ± 0.008	0.667 ± 0.008	0.669 ± 0.012
F_1^{m}	0.854 ± 0.007	0.496 ± 0.006	0.643 ± 0.042	0.514 ± 0.004	0.643 ± 0.007	0.654 ± 0.009	0.664 ± 0.012
Loops	5.1 ± 0.6	7.7 ± 1.9	4.0 ± 0.7	5.9 ± 0.9	5.0 ± 1.2	3.8 ± 0.8	4.5 ± 1.0
Updates	59.0 ± 4.3	123.4 ± 13.1	32.8 ± 4.0	32.5 ± 4.1	11.5 ± 3.4	6.5 ± 1.2	8.6 ± 2.1

TABLE II. Top rows: testing set F_1^{m} values attained by maximizing F_1^{m} with our learning algorithm (Algorithm 1), by the baseline binary relevance approach (Error rate), and by maximizing F_1^{M} (first loop of Algorithm 1). Bottom rows: number of loops of Algorithm 1 over all classes, and number binary classifiers updated. Average and standard deviation over the different runs of the experiments are reported.

toward the development of other binary learning algorithms of this kind. We then provided some empirical evidence of the effectiveness of our learning algorithm, with respect to the standard binary relevance approach, when binary classifiers that minimize the misclassification probability are used. The processing cost of our learning algorithm turned out to be just twice as the one of binary relevance.

Some interesting follow-ups of this work are the following: (i) investigating whether our results can be exploited also for implementing multilabel classifiers without using binary relevance, to take into account the correlation between labels; (ii) devising heuristics for choosing a suitable class ordering in Algorithm 1, for reducing processing cost; (iii) investigating whether incremental learning techniques for binary classifiers can be exploited for the same goal; (iv) theoretically analyzing computational complexity.

Acknowledgment

This work has been partly supported by the project "Security of pattern recognition systems in future Internet" CRP-18293 funded by Regione Autonoma della Sardegna, L.R. 7/2007, Bando 2009.

References

[1] C. J. van Rijsbergen, *Information Retrieval*, 2nd ed. London: Butterworths, 1979.

[2] D. R. Musicant, V. Kumar, and A. Ozgur, "Optimizing F-Measure with Support Vector Machines," in *Int. Florida Artificial Intelligence Research Society Conference*. AAAI Press, 2003, pp. 356–360.

[3] M. Jansche, "Maximum expected F-measure training of logistic regression models," in *Int. Conf. on Human Language Technology and Empirical Methods in Natural Language Processing*, 2005, pp. 692–699.

[4] T. Joachims, "A Support Vector Method for Multivariate Performance Measures," in *Int. Conf. on Machine Learning*, 2005, pp. 377–384.

[5] D. D. Lewis, "Evaluating and optimizing autonomous text classification systems," in *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1995, pp. 246–254.

[6] Y. Nan, K. M. A. Chai, W. S. Lee, and H. L. Chieu, "Optimizing F-measures: A tale of two approaches," in *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[7] K. Dembczynski, A. Jachnik, W. Kotlowski, W. Waegeman, and E. Hüllermeier, "Optimizing the F-measure in multi-label classification: Plug-in rule approach versus structured loss minimization," in *Proceedings of the 30th International Conference on Machine Learning*, 2013.

[8] J. Petterson and T. S. Caetano, "Submodular Multi-Label Learning," in *Advances in Neural Information Processing Systems 24*, 2011, pp. 1512–1520.

[9] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large Margin Methods for Structured and Interdependent Output Variables." *Journal of Machine Learning Research*, vol. 6, pp. 1453–1484, 2005

[10] J. Petterson and T. S. Caetano, "Reverse multi-label learning," in *Advances in Neural Information Processing Systems 23*, 2010, pp. 1912–1920.

[11] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier, "An Exact Algorithm for F-Measure Maximization," in *Neural Information Processing Systems*, 2011, pp. 1404–1412.

[12] J. R. Quevedo, O. Luaces, and A. Bahamonde, "Multilabel classifiers with a probabilistic thresholding strategy." *Pattern Recognition*, vol. 45, no. 2, pp. 876–883, 2012.

[13] I. Pillai, G. Fumera, and F. Roli, "Threshold optimisation for multi-label classifiers," *Pattern Recognition*, vol. 46, no. 7, pp. 2055–2065, 2013.

[14] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier Chains for Multi-label Classification," in *European Conf. on Machine Learning and Knowledge Discovery in Databases*, vol. 5782. Springer, 2009, pp. 254–269.

[15] Y. Yang, "A Study on Thresholding Strategies for Text Categorization," in *Int. Conf. on Research and Development in Information Retrieval*. ACM, 2001, pp. 137–145.

[16] R.-E. Fan and C.-J. Lin, "A Study on Threshold Selection for Multi-label," National Taiwan University, Tech. Rep., 2007.