A TCART-M - Tuned CARTesian-based Error Function for Multilabel Classification with the MLP

Jacek Mańdziuk
Faculty of Mathematics
and Information Science
Warsaw University of Technology,
Poland
mandziuk@mini.pw.edu.pl
and
School of Computer Science and Engineering
Nanyang Technological University, Singapore
j.mandziuk@ntu.edu.sg

Adam Żychowski
Faculty of Mathematics
and Information Science
Warsaw University of Technology,
Poland
a.zychowski@mini.pw.edu.pl

Lipo Wang
School of Electrical
and Electronic Engineering
Nanyang Technological University,
Singapore
elpwang@ntu.edu.sg

***Abstract*—In 2006 Zhang and Zhou proposed a multilabel classification model based on the MLP network, which was subsequently improved by Grodzicki et al. This paper further improves both these approaches by introducing a scaling parameter responsible for maintaining a balance between the impacts of particular components of the MLP's error function in the training process. The newly-proposed parameter is autonomously fine-tuned by the system in the nested cross validation process. The proposed approach is tested on a set of well-established benchmarks and demonstrates its superiority over the baseline methods for 16 different error measures used in the experiments. Furthermore, the method proves competitive to 12 other state-of-the-art machine learning approaches which are used for further comparisons. In the combined score composed of ranking positions for all benchmarks and all error functions, the proposed neural network system gains the leading position among all tested methods.**

I. Introduction

While the baseline formulation of a classification task consists of assigning each sample to one of the available categories (classes, labels), its multilabel version assumes that a given sample may belong to more than one category, or alternatively, be assigned more than one label from the set of all available labels. Another layer of difficulty in the multilabel classification formulation stems from the fact that the size of the assigned subset of labels is usually not known *a priori*.

Numerous real-life applications of multilabel classification (MC) led to development of various approaches to solving this problem with the used of various machine learning techniques, e.g. classifier chains [1], k-Nearest Neighbor [2], decision trees [3], random forests [4], neural networks [5], [6], and other.

In this paper, we enhance our previous approach [6] (denoted CART-M here), which relies on a specifically defined error function and operates on a Cartesian set of available labels, by introducing a *scaling parameter* D in the error function formulation. Two approaches to selection of the best value of D using nested cross validation are proposed and experimentally evaluated. The first one determines the optimal value of D by a ranking involving 16 different evaluation (loss) functions. The other one is dedicated to a particular evaluation function which is used directly to determine the best value of D.

Both the baseline version of CART-M [6] and its tuned version TCART-M proposed in this paper are thoroughly evaluated against 12 other approaches taken from a recent survey paper [7] based on 5 widely-used benchmark problems and 16 evaluation functions (leading to 80 evaluation measures altogether).

The remainder of this paper is arranged as follows. Section II presents the definition of MC and discusses its practical relevance. Section III summarizes the comparative approaches used in the experimental evaluation of the TCART-M. The next section provides a description of the TCART-M method. Section V is devoted to a presentation of the experimental setup, in particular, the benchmark sets, evaluation measures and parametrization of the methods used in the experiments. Experimental results and their comparisons with other solutions are presented in section VI. The last section is devoted to conclusions.

II. Multilabel Classification Problem

The problem of MC is defined as follows. Let $X \subseteq \mathbb{R}^d$ be d-dimensional instance space and $Y = \{y_1, y_2, \ldots, y_Q\}$ denote the set of Q possible labels. The MC task is to learn function $h : X \to 2^Y$, which to each object from the domain of instances X assigns a corresponding subset of labels.

In practical situations the above definition is often implemented as the task of finding function $f : X \times Y \to \mathbb{R}$ such that

$$\forall x_p \in X, \forall y_1 \in Y_p, y_2 \notin Y_p \qquad f(x_p, y_1) > f(x_p, y_2) \quad (1)$$

Function f provides greater outputs for the elements belonging to Y_p than for those not belonging to Y_p, where $Y_p \subseteq Y$ is a set of labels assigned to object x_p.

978-1-5090-6182-2/17/$31.00 ©2017 IEEE

MC is widely applicable to various real-life tasks, including text categorization (e.g. automatic tag suggestions for documents [8], articles [9] or e-mails [10]), multimedia files classification (e.g. music [11], video [12] or images [13] classification or tagging), or biology and bioinformatics (for instance, in discovering genomic functions [14] or finding probable diseases based on observed symptoms [15]).

III. State-of-the-Art Approaches

In recent years, a growing interest in MC has been observed mainly due to its wide applicability in various domains and, at the same time, an intrinsic complexity which makes the problem challenging in both theoretical and practical dimensions. In effect, numerous new approaches to MC relying on various machine learning techniques were proposed and proved successful in various application domains. This section presents an overview of some of the relevant MC algorithms. Please consult [16] and [7] for a more in-depth overview and the source papers for a detailed description of the respective methods.

Binary relevance (BR) [13] is one of the simplest algorithms, which relies on splitting the MC learning problem with Q classes into Q independent binary classification problems. For each of them the algorithm decides whether or not the respective label is relevant for the given example.

The *Classifier chaining* algorithm (CC) [1] also relies on Q binary classifiers but, unlike in BR, they are not pairwise independent, but chained (put in a sequence). The feature space of each classifier in a chain is extended by the results obtained for all previously lined up classifiers.

Calibrated label ranking (CLR) [17] transforms the MC learning problem into a label ranking problem. Label ranking is created by means of pairwise comparison. The method was further improved by using a more effective *Quick Weighted* voting scheme (QWML) [18].

Hierarchy Of Multilabel classifiERs (HOMER) [19] is designed mainly for large multilabel data sets and therefore carefully optimized for computational efficiency. It first groups similar labels into sets based on their distribution by means of a balanced clustering algorithm similar to the k means algorithm. Afterwards, these smaller sets are considered separately.

Multi-Label C4.5 (ML-4.5) [20] is an adaptation of a popular C4.5 classification tree algorithm with an appropriately modified entropy formula and multiple labels (sets of labels) stored in the leaves of a classification tree.

Multilabel k-nearest neighbors (ML-kNN) [2] is another example of adaptation of a very popular classification algorithm to a multilabel version. For each new object, its k nearest neighbors among objects with known label sets are firstly identified, and then, based on these label sets, the maximum a posteriori principle is used to determine the set of labels for the considered object.

Predictive clustering trees (PCT) [3] take a hierarchical clustering approach - a root node represents one cluster with all the data and down the tree the data is partitioned into smaller clusters. The method constructs a tree using a standard top-down induction of decision trees by maximizing the cluster's variance reduction. The prototype function returns a vector of probabilities that an instance is labeled with a given label.

RAndom k-labELsets (RAkEL) [21] uses an ensemble approach. It randomly creates subsets composed of k labels and uses them to train a label power-set classifier, which considers each distinct combination of labels that exist in the training set as a different class. Final prediction for a particular label is obtained by a simple voting scheme based on decisions of classifiers that contain this label.

Ensembles of classifier chains (ECC) [1] method uses multiple instances of the CC algorithm (described above) with random labels in a chain order and random subsets of training samples. Predictions from all instances are ultimately summed up to yield the final result.

Random forest of ML-C4.5 (RFML-C4.5) [4] and *Random forest of predictive clustering trees* (RF-PCT) [7] are ensemble methods that use random forests with ML-C4.5 and PCT baseline classifiers, respectively. The predictions of baseline classifiers are combined using one of standard voting schemes.

Generally speaking, all the above-mentioned methods can be divided into 3 groups: *problem transformation methods*, *algorithm adaptation methods* and *ensemble methods*. The first category (which encompasses BR, CC, CLR, QWML and HOMER methods) refers to algorithms which transform the problem of MC into another well-studied machine learning scenario. The algorithm adaptation methods (which include ML-C4.5, PCT and ML-kNN approaches) adapt one of the popular machine learning techniques to dealing with MC data sets and MC problem formulation. The final group (i.e. RAkEL, ECC, RF-MLC4.5 and RF-PCT methods) are ensemble approaches which rely on independent running of multiple instances of simple classifiers belonging to one of the two previous groups on modified data sets and combining the results based on some voting scheme. According to [7] the latter group contains the most powerful approaches which are generally superb over the single-classifier approaches. Certainly these ensemble methods are also the most complex in terms of parametrization and time requirements.

Surprisingly, there are not so many approaches to MC employing neural networks. Probably the most popular one is the *Backpropagation for Multilabel Learning* (BP-MLL) [5] which relies on using a one-hidden layer perceptron with the input layer equal to the number of attributes in a considered data set with an additional bias neuron and the output layer equal to the number of labels. Training is preformed with the classic backpropagation algorithm with the error function of the following form:

$$E_{BP-MLL} = \sum_{p=1}^{m} \frac{\sum_{(r,s)\in Y_p \times \overline{Y}_p} e^{-(c_r^p - c_s^p)}}{|Y_p||\overline{Y_p}|} \qquad (2)$$

where $Y_p \subseteq Y$ is a set of labels assigned to the p-th training sample x_p, $\overline{Y_p}$ is a complementary set to Y_p (i.e. $\overline{Y_p} = Y \setminus Y_p$), c_q^p is current output of the neuron associated with the q-th

label, m is the number of training instances. The particular form of the error function (2) is intended to provide greater outputs of neurons corresponding to the labels belonging to Y_p compared to the remaining ones (not from Y_p). The set of labels assigned to a given input sample x_p is defined by the set of neurons whose outputs exceed a certain pre-defined, input-dependent threshold $t(x_p)$.

The BP-MLL approach was subsequently improved in [6] by extending the form of the error function in a twofold way. First of all, the values of (previously pre-defined) threshold $t(x_p)$ were included in the network's error function and made independent directly of the input instance (they were assigned to the output nodes, i.e. particular labels). Second of all, in the error function comparisons between all c_q^p values of categories belonging to Y_p (and to $\overline{Y_p}$, respectively) were taken into account. These two modifications led to the following form of the error function [6]:

$$E_{CART-M} = \sum_{p=1}^{m} \left(\frac{\sum_{(r,s)\in Y_p \times \overline{Y}_p} (e^{-(c_{2r}^p - c_{2s}^p)} + e^{-(c_{2s+1}^p - c_{2r+1}^p)})}{2|Y_p||\overline{Y}_p| + |Y_p|^2 + |\overline{Y}_p|^2} + \frac{\sum_{r\in Y_p}\sum_{t\in Y_p} e^{-(c_{2r}^p - c_{2t+1}^p)} + \sum_{s\in \overline{Y}_p}\sum_{t\in \overline{Y}_p} e^{-(c_{2t+1}^p - c_{2s}^p)}}{2|Y_p||\overline{Y}_p| + |Y_p|^2 + |\overline{Y}_p|^2} \right) \quad (3)$$

where for each category (each label) i the two neurons indexed by $2i$ and $2i+1$, respectively, represent the output of the neuron corresponding to the ith label and the threshold value associated with this output neuron (ith label).

IV. Proposed Modified Approach

The above-described CART-M neural network multilabel classifier was tested in [6] on the *yeast genome* data set [14] (which is one of the standard benchmark problems in MC) with the use of 3 evaluation (loss) functions. The results appeared to be statistically significantly better than those of the original BP-MLL model [5] and slightly better (with no statistical relevance) than ADTBOOST.HM [22] and RANK-SVM [14] models, being ones of the strongest approaches at that time.

In this paper we propose further modification of the BP-MLL and CART-M methods relying on further tuning of the global error function of the MLP network. While the CART-M model with the error function (3) clearly outperforms the original formulation of BP-MLL, in the recent in-depth testing of the CART-M statistical properties, it turned out that the error function of the model is biased towards the newly-added components at the expense of the baseline formulation focusing on the $e^{-(c_{2r}^p - c_{2s}^p)}$ component. For this reason we investigated the possibility of adding a scaling parameter D to the error function formulation and its automatic tuning for a given data set. In this respect, the tuned error function (denoted by TCART-M) has the following form:

$$E_{TCART-M} = \sum_{p=1}^{m} \left(\frac{\sum_{(r,s)\in Y_p \times \overline{Y}_p} (e^{-(c_{2r}^p - c_{2s}^p)} + \frac{e^{-(c_{2s+1}^p - c_{2r+1}^p)}}{D})}{2|Y_p||\overline{Y}_p| + |Y_p|^2 + |\overline{Y}_p|^2} + \frac{\sum_{r\in Y_p}\sum_{t\in Y_p} \frac{e^{-(c_{2r}^p - c_{2t+1}^p)}}{D} + \sum_{s\in \overline{Y}_p}\sum_{t\in \overline{Y}_p} \frac{e^{-(c_{2t+1}^p - c_{2s}^p)}}{D}}{2|Y_p||\overline{Y}_p| + |Y_p|^2 + |\overline{Y}_p|^2} \right) \quad (4)$$

Numerical results presented in section VI fully confirmed the above-mentioned reasoning. In the extensive tests on 5 benchmark sets with 16 independently measured error (loss) functions our new TCART-M method appears not only to be stronger than CATR-MC and BP-MLL, but is also comparable or better (in average) than the competitive state-of-the-art approaches which use various machine learning techniques as presented in section III.

V. Experimental Setup

The proposed TCART-M method was compared with 14 different approaches briefly described in Section III: BR [13], CC [1], CLR [17], QWML [18], HOMER [19], ML-C4.5 [20], PCT [3], ML-kNN [2], RAkEL [21], ECC [1], FML-C4.5 [4], RF-PCT [7], BP-MLL [5] and CART-M [6]. The results for the first 12 methods were obtained from an experimental-based survey paper [7] and the remaining two approaches (BP-MLL and CART-M) were implemented by the authors according to their descriptions presented in [5] and [6], respectively.

A. Parameter selection

In order to make comparison fair, all parameters of the proposed modified TCART-M method were set according to the original selection proposed in [6]. In particular, the hidden layer size was not optimized and contained 40 neurons. The input layer was equal to the number of attributes, plus an additional bias neuron, and the output layer was composed of $2Q$ neurons, where Q is the number of possible labels. Following [6], the learning rate was set to 0.05, the weight decay to 0.5 and the number of training epochs was equal to 100. In all experiments, a 10-fold cross validation (CV) was applied.

The value of a scaling parameter D in (4) was chosen using a nested CV technique among the following 16 candidate values: $D_{cand} = \{0.25, 0.5, \ldots, 1.75, 2\} \cup \{2.5, 3, 3.5, 4, 5, 6, 8, 10\}$. Two different measures of efficiency of a given selection of D in the inner (nested) CV loop were proposed and tested in the experiment: g-general and *i*-individual.

g: the optimal value of D is determined based on the ranking which involves all 16 evaluation measures. More precisely, for each error measure, the scores for all 16 candidate values $d_i \in D_{cand}$ are calculated and ranked by sorting in the descending order starting from the best one. Afterwards,

for each d_i the positions in all rankings (for all 16 error measures) are summed up and the one with the lowest result is selected as D.

i: for a given error measure (one of 16 available) the optimal value of D is chosen directly based on comparison of results obtained for this pre-selected evaluation measure.

The former evaluation method (TCART-M*g*) is universal and independent of the choice of an evaluation measure that may be used in practical situations later on. The latter approach (TCART-M*i*) is dedicated to a particular error measure and optimized for the subsequent method's use with that exact measure (though, "by chance", may be well tuned for other measures, as well).

B. Benchmark problems

The method was tested on 5 widely-used multilabel classification benchmark problems. These data sets have various characteristics and, in particular, differ in the numbers of instances, numbers of labels and average labels' cardinalities. The data come from three real-life domains: text categorization, multimedia (images and music) categorization and biology. Table I summarizes the basic parameters of the used data sets. The *Yeast* data set [14] is one of the most popular

Name	Domain	Instances	Attributes	Labels	Card.
yeast	biology	2417	103	14	4.24
scene	multimedia	2407	294	6	1.07
emotions	multimedia	593	72	6	1.87
enron	text	1702	1001	53	3.38
medical	text	978	1449	45	1.25

TABLE I
BASIC PARAMETERS OF THE USED BENCHMARK DATA SETS: DOMAIN, NUMBER OF INSTANCES, NUMBER OF ATTRIBUTES, NUMBER OF LABELS AND AVERAGE CARDINALITY OF THE LABELS.

benchmarks in the MC domain. The data represents genes, each of which can be associated with a selection of 14 biological functions (labels).

Scene data set [13] instances are images of landscapes which should be annotated with the subsets of the following set of labels: {mountain, beach, field, fall-foliage, sunset, urban}.

Emotions [11] is a data set containing music extracts. The goal is to annotate each of them with a subset of the following six emotions: {happy-pleased, angry-aggressive, sad-lonely, amazed-surprised, quiet-still, relaxing-calm}.

The *Enron* data set [10] is related to e-mail categorization. Objects are e-mails from Enron Corporation written by 150 people and marked by 53 labels. The labels are grouped into 4 main categories: coarse genre, forwarded messages, primary subjects, and messages with emotional tone.

Finally, the *medical* data set [15] is composed of pieces of text which briefly describe patient's symptoms history. The labels that need to assigned are potential diseases listed in the International Classification of Diseases [23].

All 5 the above-mentioned benchmark problems are very popular within the multilabel classification research community and often used for comparing different MC approaches. In particular, they were used in [7] which is the source of comparative results used in this paper to assess the quality of the proposed TCART-M algorithm.

C. Evaluation measures

In order to make the comparison fair and as thorough as possible, we partly follow the experimental setup proposed in [7] and, instead of applying one particular error measure, propose the use of an ensemble of 16 error functions. These error measures can be roughly divided into three groups. For the sake of space limits we will restrict our description to one example measure per group. Please refer to [7] or any machine learning book for the definitions and interpretations of the remaining ones. In the following description, N denotes the number of test instances, x_i is the ith test instance and Y_i is the true set of labels assigned to this sample.

The first group includes *example-based measures*, which rely on calculating the difference between the actual and predicted values separately for each test sample and then calculating the average value across the whole test set. The most common example-based measure is *precision* defined as:

$$precision(h) = \frac{1}{N}\sum_{i=1}^{n}\frac{|h(x_i) \cap Y_i|}{|Y_i|} \tag{5}$$

where $h(x_i)$ denotes the set of labels assigned to x_i by the classifier being assessed. *Precision* can be interpreted as the average (across the whole test set) fraction of correctly assigned labels. In the experiments, 6 *example-based measures* were used: *Hamming loss*, *accuracy*, *precision*, *recall*, *F1 score* and *subset accuracy*.

Error measures in the second group are calculated by first evaluating the classifier's performance separately on each label, and then returning the mean value across all labels. Therefore, these are the *label-based measures*. One of the representatives of this category is *macro precision* defined by the following formula:

$$macro\_precision = \frac{1}{Q}\sum_{j=1}^{Q}\frac{tp_j}{tp_j + fp_j} \tag{6}$$

where Q denotes the number of all possible labels, tp_j and fp_j are respectively the numbers of true positives and false positives obtained by the classifier for label j across the whole test set. Intuitively, *macro-precision* measures the ability of a classifier to not assign a label in the cases in which it should actually not be assigned. There are 6 *label-based measures* used in the experiments: *micro-precision*, *micro-recall*, *micro-*F_1, *macro-precision*, *macro-recall* and *macro-*F_1.

The third group is composed of *ranking-based measures*, which for each tested sample build a ranking of labels which are presumably the best suited for this sample. This ranking is compared with a ground-truth ranking and certain aspects of these comparisons across all test samples are measured and

	BP-MLL	CART-M	TCART-M*g*	TCART-M*i*
Hamming Loss	0.203	0.201	0.187	**0.186**
Accuracy	0.570	0.547	0.579	**0.580**
Precision	0.652	0.666	0.682	**0.683**
Recall	**0.730**	0.663	0.700	0.703
Subset Accuracy	0.299	0.291	0.331	**0.332**
F1 score	0.660	0.632	0.660	**0.663**
Micro-precision	0.657	0.685	**0.701**	0.695
Macro-precision	0.657	0.685	**0.700**	0.696
Micro-recall	**0.728**	0.658	0.698	0.706
Macro-recall	**0.714**	0.650	0.686	0.703
Micro-F1	0.690	0.671	**0.699**	0.693
Macro-F1	0.676	0.658	0.685	**0.690**
Ranking Loss	0.160	0.160	0.145	**0.142**
OneError	0.290	0.275	**0.251**	0.253
Coverage	1.748	1.766	1.694	**1.658**
Average Precision	0.799	0.803	**0.818**	0.812

TABLE II
COMPARISON OF RESULTS FOR THE *emotions* BENCHMARK SET (AVERAGED OVER 30 TRIALS). THE BEST RESULTS FOR EACH EVALUATION MEASURE ARE BOLDED.

	BP-MLL	CART-M	TCART-M*g*	TCART-M*i*
Hamming Loss	0.217	**0.197**	0.198	0.198
Accuracy	**0.535**	0.510	0.510	0.509
Precision	0.637	0.700	**0.706**	0.704
Recall	**0.711**	0.595	0.596	0.601
Subset Accuracy	0.143	**0.168**	0.160	0.161
F1 score	0.017	0.600	0.618	**0.621**
Micro-precision	0.626	**0.709**	**0.709**	0.706
Macro-precision	0.464	**0.544**	0.533	0.541
Micro-recall	**0.702**	0.592	0.590	0.590
Macro-recall	**0.468**	0.374	0.359	0.369
Micro-F1	**0.661**	0.645	0.644	0.643
Macro-F1	**0.441**	0.405	0.383	0.406
Ranking Loss	0.174	**0.164**	0.165	**0.164**
OneError	0.237	0.227	0.225	**0.224**
Coverage	6.441	6.285	6.305	**6.265**
Average Precision	0.754	**0.767**	0.766	**0.767**

TABLE III
COMPARISON OF RESULTS FOR THE *yeast* BENCHMARK SET (AVERAGED OVER 30 TRIALS). THE BEST RESULTS FOR EACH EVALUATION MEASURE ARE BOLDED.

evaluated. As an example let's look at the *average precision* measure defined by the following formula:

$$average\_precision(f) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{|Y_i|}\sum_{q\in Y_i}\frac{|L_i(q)|}{rank_f(x_i,q)} \quad (7)$$

where f is a prediction function and $rank_f$ is the f-based ranking defined in a way that for any labels q_1, q_2 if $f(x_i, q_1) > f(x_i, q_2)$ then $rank_f(x_i, q_1) < rank_f(x_i, q_2)$. $L_i(q) = \{q' \in Y_i : rank_f(x_i, q') \leq rank_f(x_i, q)\}$ is a set of labels from Y_i that are ranked above a given label $q \in Y_i$. The *average_precision* measures the average fraction of the labels that are ranked above a given label $q \in Y_i$ that, in fact, are in Y_i. Four *ranking-based measures* were applied in the experiments, namely *average precision*, *one-error*, *coverage* and *ranking loss*.

15 measures (all except *coverage*) yield values within the interval $[0, 1]$. In the case of *Hamming loss*, *one-error*, *coverage* and *ranking loss*, the smaller the value, the better the method's performance. For all the remaining metrics, the greater the value, the better the performance.

VI. EXPERIMENTAL RESULTS

The experiments were arranged along the two main goals: (1) comparison of the proposed modified method TCART-M*g/i* with the methods that inspired the introduced modifications, i.e. BP-MLL and CART-M, and (2) comparison of the neural network-based methods (the ones mentioned in (1) above) with other machine learning approaches.

For the purpose of neural network models comparisons, 30 independent tests were run for each model, each benchmark problem, and each of the 16 error functions.

Tables II and III show the results obtained for *emotions* and *yeast* benchmarks, which appeared to be the two extreme cases from the point of view of the proposed TCART-M method.

In case of *yeast* benchmark (which appeared to be the least fitted to the proposed modifications) the modified methods TCART-M*g* and TCART-M*i* accomplished the results slightly inferior to the baseline formulation, being in average weaker than CART-M by 0.78% and 0.06%, respectively.

On the other hand, for the *emotions* benchmark, for all 16 evaluations measures, the mean results of both modified approaches outperformed the baseline method by a clear margin - the average improvement across all measures is equal to 5.45% and 5.91% for TCART-M*g* and TCART-M*i*, respectively. For the remaining 3 data sets (*scene*, *enron*, *medical*), the improvement of the proposed system is also clear, though not as striking as in the case of the *emotions* data set. The summary of results is presented in Table IV. All in all, out of 80 tested cases the TCART-M*g* and TCART-M*i* appeared to be superior to CART-M in 55 and 60 of them, respectively. Based on 1-tailed t-test with significance level equal to 0.05 respectively 42 and 47 of these results are statistically significant. For the sake of space limits detailed p-values are not presented. All of them are in the range of 0.0001 to 0.26. Normal distribution of data was checked by Shapiro-Wilk test.

In the other set of experiments, a comparison between neural network approaches and the ones listed in section III was performed based on average ranking positions. Namely, for a given benchmark set, for each tested algorithm, separate rankings for all error measures were constructed and then positions in these rankings were summed up. The best possible sum of positions was 16 (i.e. the 1st position for each of 16 evaluation measures), the worst possible outcome was equal to 256 (the 16th position in each of the 16 measure-related rankings). The summary of results is presented in Table V. It can be seen in the table that, for the *emotions* data set, the proposed methods are the best ones with more than 30 points ahead of the remaining methods. Also for the *enron* bench-

	TCART-M*g*		TCART-M*i*	
Benchmark	Mean	D	Mean	D
emotions	16 (16)	6.7	16 (16)	6.1
yeast	4 (2)	2.0	7 (4)	2.5
scene	12 (9)	5.1	12 (11)	6.3
enron	11 (8)	4.7	13 (9)	4.5
medical	12 (7)	2.0	12 (7)	2.1

TABLE IV
FOR EACH METHOD, THE LEFT COLUMN PRESENTS A NUMBER OF EVALUATION MEASURES (OUT OF 16) FOR WHICH PROPOSED MODIFICATIONS (RESPECTIVELY TCART-M*g* AND TCART-M*i*) YIELDED BETTER MEAN RESULTS THAN CART-M. THE VALUE IN PARENTHESES DENOTES THE NUMBER OF STATISTICALLY SIGNIFICANT RESULTS (1-TAILED T-TEST WITH SIGNIFICANCE LEVEL EQUAL TO 0.05). FOR BOTH METHODS, THE RIGHT COLUMN PRESENTS THE AVERAGE VALUE OF D SELECTED BY NESTED CV (CART-M METHOD CORRESPONDS TO THE CASE OF $D = 1$).

mark, both modifications are placed in the top-3 positions. The worst performance can be observed for the *medical* data set where they are located at 10th and 11th position, respectively. However, **in the overall measure obtained by summing the ranking points over all** 5 **benchmarks and all** 16 **error functions, the proposed algorithms gained the top two positions.** The detailed results regarding the average values of particular measures of the tested methods are presented in Tables VI, VII, VIII, IX and X - each devoted to one benchmark problem.

One of the key aspects of the proposed method is introduction of scaling parameter D and its tuning by means of a nested CV procedure. The average values assigned to this parameter and its standard deviations across all 30 runs are shown in Table IV. Analysis of results suggests that there is some correlation between the range of achieved improvement (compared to CART-M) and selection of D. For the two data sets, *yeast* and *medical* which were the least favorable for our method D values are small, between 1.0 and 3.0. For the remaining benchmarks D is usually between 4.0 and 6.0 or even above 6.0 for the *emotions* data set (please note that this is the set for which the striking improvement in all 16 measures was achieved). Moreover, the situations in which D was selected lower than 1.0 were not noticed (even though the values of 0.25, 0.5 and 0.75 were available for selection). Furthermore, only occasionally $D = 1.0$ was chosen (in that case TCART-M becomes equivalent to CART-M). The above observations confirm our intuition that the impact of the new components in the error function (in CART-M) is too strong and should be reduced (divided by $D > 1.0$). Another conclusion stemming from the results is the advantage of the TCART-M*i* over TCART-M*g* variant, which could be, to a large extent, expected as unlike the latter approach, which is optimized for a particular error measure, the former method takes a much universal approach with no pre-defined error measure in mind. However, despite its advantage, the TCART-M*i* is not necessarily the best option in practice, in particular, when the error measure to be used is not known in advance. In such cases the value of D fine-tuned for a certain error measure may prove ineffective when the assessment function is changed.

VII. CONCLUSIONS

In 2006 Zhang and Zhou [5] proposed a neural network-based approach to the multilabel classification problem. The proposed model was later improved by Grodzicki et al. [6] thanks to restructuring the error function form by means of adding specific cartesian-like (pair-based) components. This paper further improves the model by introducing a scaling parameter which is autonomously fine-tuned by the system in the nested cross validation process. Two ways of measuring the efficacy of this internal scaling parameters in the inner (nested) CV loop are proposed, referred to as g (general) and i (individual). Both variants of the proposed method were tested on a set of 5 well-established benchmarks and proven for an ensemble of 16 different error measures. The experimental results confirmed that both new approaches (TCART-M*g* and TCART-M*i*) in most of the cases outperform the previous versions proposed in [5] and [6].

Except for the above-described tests, the proposed methods were also experimentally compared with 12 popular multilabel classification methods based on the results published in a recent survey [7]. The results are very promising, in several combinations of a benchmark set and error measure the newly-proposed neural models are clearly the most efficient, in many other cases, they are in the top selection. In the combined score obtained by summing the ranking positions in all 5 benchmarks and 16 evaluation measures, the proposed algorithms are located in the first two places. While definitely further comparisons and more in-depth investigations are necessary to fully confirm the strength of the proposed models, based on the hitherto results, it is safe to say that the neural network based TCART-M method introduced in this paper is a viable alternative to other state-of-the-art machine learning approaches.

REFERENCES

[1] J. Read, B. Pfahringer, G. Holmes, and E. Frank, *Classifier Chains for Multi-label Classification*. Springer Berlin Heidelberg, 2009, pp. 254–269.

[2] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038 – 2048, 2007.

[3] H. Blockeel, L. De Raedt, and J. Ramon, "Top-down induction of clustering trees," *arXiv preprint cs/0011032*, 2000.

[4] D. Kocev, C. Vens, J. Struyf, and S. Džeroski, "Ensembles of multi-objective decision trees," in *European Conference on Machine Learning*. Springer, 2007, pp. 624–631.

[5] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. on Knowl. and Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[6] R. Grodzicki, J. Mańdziuk, and L. Wang, "Improved Multilabel Classification with Neural Networks," in *Parallel Problem Solving from Nature*, ser. Lecture Notes in Computer Science, vol. 5199. Springer Verlag, 2008, pp. 409–416.

[7] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Deroski, "An extensive experimental comparison of methods for multi-label learning," *Pattern Recognition*, vol. 45, no. 9, pp. 3084 – 3104, 2012, best Papers of Iberian Conference on Pattern Recognition and Image Analysis (IbPRIA'2011).

[8] A. N. Srivastava and B. Zane-Ulman, "Discovering recurring anomalies in text reports regarding complex space systems," in *IEEE Aerospace Conference.*, 2005, p. 5563.

emotions	yeast	scene	enron	medical	sum
TCART-M*i* (34)	CLR (72)	BR (78)	**TCART-M*i*** (74)	QWML (76)	**TCART-M*i*** (942)
TCART-M*g* (41)	BR (99)	CC (79)	BR (84)	HOMER (83)	**TCART-M*g*** (1026)
BP-MLL (72)	HOMER (110)	RAkEL (79)	**TCART-M*g*** (86)	CLR (90)	CLR (1056)
RF-PCT (72)	**TCART-M*i*** (115)	ECC (90)	CLR (89)	ML-C4,5 (94)	BR (1152)
CART-M (79)	CC (115)	CLR (92)	CART-M (106)	RF-PCT (106)	HOMER (1166)
RFML-C4,5 (90)	CART-M (117)	**TCART-M*i*** (103)	RF-PCT (110)	RAkEL (120)	RF-PCT (1166)
ML-C4,5 (102)	QWML (123)	**TCART-M*g*** (106)	HOMER (119)	CC (124)	CART-M (1176)
PCT (154)	**TCART-M*g*** (132)	HOMER (107)	ECC (127)	BR (137)	CC (1306)
HOMER (164)	ECC (132)	CART-M (124)	RFML-C4,5 (137)	ECC (137)	RAkEL (1338)
BR (178)	RAkEL (135)	QWML (133)	CC (146)	**TCART-M*i*** (145)	ECC (1350)
RAkEL (184)	BP-MLL (138)	RF-PCT (155)	RAkEL (151)	**TCART-M*g*** (148)	QWML (1380)
CLR (185)	RF-PCT (140)	ML-k NN (182)	QWML (165)	ML-k NN (149)	RFML-C4,5 (1536)
CC (189)	ML-k NN (145)	RFML-C4,5 (185)	ML-C4,5 (171)	CART-M (162)	ML-C4,5 (1554)
ECC (189)	RFML-C4,5 (180)	BP-MLL (208)	BP-MLL (174)	RFML-C4,5 (176)	BP-MLL (1586)
QWML (193)	ML-C4,5 (194)	ML-C4,5 (216)	ML-k NN (195)	BP-MLL (201)	ML-k NN (1826)
ML-k NN (242)	PCT (218)	PCT (233)	PCT (232)	PCT (215)	PCT (2104)

TABLE V

METHODS SORTED BY RANKING SUMMED ON EVALUATION MEASURES DIVIDED FOR BENCHMARK PROBLEMS. VALUES IN BRACKETS ARE SUM OF METHOD'S POSITIONS ACROSS EVALUATION MEASURES. PROPOSED MODIFICATIONS ARE BOLDED.

	BP-MLL	CART-M	TCART-M*g*	TCART-M*i*	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-k NN	RAkEL	ECC	RFML-C4.5	RF-PCT
Hamming Loss	0.203	0.201	0.187	**0.186**	0.257	0.256	0.257	0.254	0.361	0.247	0.267	0.294	0.282	0.281	0.198	0.189
Accuracy	0.570	0.547	0.579	**0.580**	0.361	0.356	0.361	0.373	0.471	0.536	0.448	0.319	0.419	0.432	0.488	0.519
Precision	0.652	0.666	0.682	**0.683**	0.550	0.551	0.538	0.548	0.509	0.606	0.577	0.502	0.564	0.580	0.625	0.644
Recall	0.730	0.663	0.700	0.704	0.409	0.397	0.410	0.429	**0.775**	0.703	0.534	0.377	0.491	0.533	0.545	0.582
Subset Accuracy	0.299	0.291	0.331	**0.332**	0.129	0.124	0.144	0.149	0.163	0.277	0.223	0.084	0.208	0.168	0.272	0.307
F1 score	0.660	0.632	0.660	**0.663**	0.469	0.461	0.465	0.481	0.614	0.651	0.554	0.431	0.525	0.556	0.583	0.611
Micro-precision	0.657	0.685	0.701	0.695	0.684	0.698	0.685	0.680	0.471	0.607	0.607	0.584	0.586	0.579	**0.783**	**0.783**
Macro-precision	0.657	0.685	0.700	0.697	0.721	0.581	0.677	0.660	0.464	0.602	0.628	0.518	0.547	0.531	**0.828**	0.802
Micro-recall	0.728	0.658	0.698	0.706	0.406	0.393	0.409	0.431	**0.782**	0.712	0.539	0.376	0.489	0.531	0.551	0.589
Macro-recall	0.714	0.650	0.686	0.703	0.378	0.364	0.381	0.398	**0.775**	0.702	0.533	0.334	0.462	0.508	0.532	0.569
Micro-F1	0.690	0.671	**0.699**	0.693	0.509	0.503	0.512	0.528	0.588	0.655	0.571	0.457	0.533	0.554	0.647	0.672
Macro-F1	0.676	0.658	0.684	**0.690**	0.440	0.420	0.443	0.458	0.570	0.630	0.568	0.385	0.488	0.500	0.620	0.650
Ranking Loss	0.160	0.160	0.145	**0.142**	0.246	0.245	0.264	0.331	0.297	0.210	0.270	0.283	0.281	0.310	0.153	0.151
OneError	0.290	0.275	**0.251**	0.253	0.386	0.376	0.391	0.391	0.411	0.347	0.386	0.406	0.396	0.426	0.277	0.262
Coverage	1.748	1.766	1.694	**1.658**	2.307	2.317	2.376	2.807	2.634	2.069	2.356	2.490	2.465	2.619	1.801	1.827
Average Precision	0.799	0.803	**0.818**	0.812	0.721	0.724	0.718	0.679	0.698	0.759	0.713	0.694	0.713	0.687	0.812	0.812

TABLE VI

THE AVERAGE RESULTS FOR THE *emotions* BENCHMARK SET. BEST RESULTS FOR EACH EVALUATION MEASURE ARE BOLDED.

	BP-MLL	CART-M	TCART-M*g*	TCART-M*i*	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-k NN	RAkEL	ECC	RFML-C4.5	RF-PCT
Hamming Loss	0.217	0.197	0.198	0.198	**0.190**	0.193	**0.190**	0.191	0.207	0.234	0.219	0.198	0.192	0.207	0.205	0.197
Accuracy	0.535	0.510	0.509	0.508	0.520	0.527	0.524	0.523	**0.559**	0.480	0.440	0.492	0.531	0.546	0.453	0.478
Precision	0.637	0.700	0.706	0.704	0.722	0.727	0.719	0.718	0.663	0.620	0.705	0.732	0.715	0.667	0.738	**0.744**
Recall	0.711	0.595	0.596	0.602	0.591	0.600	0.601	0.600	**0.714**	0.608	0.490	0.549	0.615	0.673	0.491	0.523
Subset Accuracy	0.143	0.168	0.160	0.161	0.190	**0.239**	0.195	0.192	0.213	0.158	0.152	0.159	0.201	0.215	0.129	0.152
F1 score	0.017	0.600	0.618	0.621	0.650	0.657	0.655	0.654	**0.687**	0.614	0.578	0.628	0.661	0.670	0.589	0.614
Micro-precision	0.626	0.709	0.709	0.707	0.733	0.726	0.729	0.727	0.647	0.618	0.698	0.736	0.720	0.662	0.747	**0.755**
Macro-precision	0.464	0.544	0.533	0.541	0.628	0.602	0.614	0.614	0.471	0.377	0.479	0.600	0.480	0.391	0.533	**0.674**
Micro-recall	**0.702**	0.592	0.590	0.590	0.587	0.588	0.595	0.595	**0.702**	0.603	0.492	0.543	0.602	0.655	0.491	0.521
Macro-recall	**0.468**	0.374	0.359	0.369	0.355	0.357	0.361	0.361	0.466	0.375	0.269	0.308	0.352	0.388	0.257	0.286
Micro-F1	0.661	0.645	0.644	0.643	0.652	0.650	0.655	0.654	**0.673**	0.610	0.577	0.625	0.656	0.658	0.593	0.617
Macro-F1	0.441	0.405	0.383	0.406	0.392	0.390	0.392	0.394	**0.447**	0.370	0.293	0.336	0.359	0.350	0.283	0.322
Ranking Loss	0.174	0.164	0.165	0.164	0.164	0.170	**0.163**	0.296	0.205	0.225	0.199	0.172	0.259	0.224	0.173	0.167
OneError	0.237	0.227	0.225	**0.224**	0.236	0.268	0.229	0.233	0.248	0.312	0.264	0.234	0.254	0.249	0.250	0.248
Coverage	6.441	6.285	6.305	6.272	6.330	6.439	6.286	8.659	7.285	7.105	6.705	6.414	7.983	7.153	6.276	**6.179**
Average Precision	0.754	0.767	0.766	0.767	**0.768**	0.755	**0.768**	0.698	0.740	0.706	0.724	0.758	0.715	0.734	0.749	0.757

TABLE VII

THE AVERAGE RESULTS FOR THE *yeast* BENCHMARK SET. BEST RESULTS FOR EACH EVALUATION MEASURE ARE BOLDED.

[9] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *Proceedings of the ECML/PKDD 2008 Discovery Challenge*, 2008.

[10] B. Klimt and Y. Yang, *The Enron Corpus: A New Dataset for Email Classification Research*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 217–226.

[11] K. Trohidis, G. Tsoumakas, G. Kalliris, and I. Vlahavas, "Multi-label classification of music into emotions," in *Proceedings of the 9th International Conference on Music Information Retrieval*, Philadelphia, USA, September 14-18 2008, pp. 325–330.

[12] C. G. M. Snoek, M. Worring, J. C. van Gemert, J. M. Geusebroek, and A. W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *ACM International Conference on Multimedia*, 2006, pp. 421–430.

[13] M. Boutell, J. Luo, X. Shen, and C. Brown, "Learning multi-label scene classification," *Pattern recognition*, vol. 37, no. 9, pp. 1757–1771, 2004.

[14] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *In Advances in Neural Information Processing Systems 14*. MIT Press, 2001, pp. 681–687.

[15] J. P. Pestian, C. Brew, P. Matykiewicz, D. J. Hovermale, N. Johnson, K. B. Cohen, and W. Duch, "A shared task involving multi-label classification of clinical free text," in *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, ser. BioNLP '07. Stroudsburg, PA, USA: Association for Computational Linguistics, 2007, pp. 97–104.

[16] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algo-

	BP-MLL	CART-M	TCART-M*g*	TCART-M*i*	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-k NN	RAkEL	ECC	RFML-C4.5	RF-PCT
Hamming Loss	0.272	0.090	0.090	0.089	0.079	0.082	0.080	0.081	0.082	0.141	0.129	0.099	**0.077**	0.085	0.116	0.094
Accuracy	0.357	0.695	0.705	0.695	0.689	0.723	0.686	0.683	0.717	0.569	0.538	0.629	0.734	**0.735**	0.388	0.541
Precision	0.361	0.715	0.724	0.728	0.718	0.758	0.714	0.711	0.746	0.592	0.565	0.661	0.768	**0.770**	0.403	0.565
Recall	**0.799**	0.735	0.768	0.774	0.711	0.726	0.712	0.709	0.744	0.582	0.539	0.655	0.740	0.771	0.388	0.541
Subset Accuracy	0.058	0.636	0.625	0.627	0.639	0.685	0.633	0.630	0.661	0.533	0.509	0.573	**0.694**	0.665	0.372	0.518
F1 score	0.479	0.715	**0.732**	0.730	0.639	0.685	0.633	0.630	0.661	0.533	0.509	0.573	0.694	0.665	0.372	0.518
Micro-precision	0.387	0.762	0.744	0.759	0.843	0.814	0.835	0.832	0.804	0.619	0.512	0.691	0.831	0.773	**0.960**	0.930
Macro-precision	0.437	0.780	0.770	0.780	0.844	0.817	0.835	0.832	0.807	0.635	0.682	0.784	0.835	0.785	**0.963**	0.919
Micro-recall	**0.796**	0.725	0.758	0.771	0.694	0.708	0.695	0.692	0.727	0.570	0.521	0.634	0.721	0.751	0.572	0.523
Macro-recall	**0.796**	0.734	0.765	0.777	0.703	0.716	0.704	0.701	0.734	0.573	0.529	0.647	0.727	0.757	0.381	0.533
Micro-F1	0.516	0.742	0.750	0.748	0.761	0.757	0.758	0.756	0.764	0.593	0.516	0.661	**0.772**	0.762	0.717	0.669
Macro-F1	0.539	0.749	0.760	0.750	0.765	0.762	0.762	0.759	0.768	0.596	0.593	0.692	**0.777**	0.770	0.514	0.658
Ranking Loss	0.180	0.076	0.072	0.074	**0.060**	0.064	0.065	0.103	0.119	0.169	0.174	0.093	0.104	0.103	0.079	0.072
OneError	0.560	0.231	0.229	0.229	**0.180**	0.204	0.190	0.193	0.216	0.394	0.389	0.242	0.197	0.213	0.232	0.210
Coverage	0.982	0.464	0.442	0.430	**0.399**	0.417	0.423	0.631	0.739	0.945	0.964	0.569	0.635	0.625	0.495	0.461
Average Precision	0.675	0.864	0.867	0.866	**0.893**	0.881	0.886	0.864	0.848	0.751	0.745	0.851	0.862	0.856	0.862	0.874

TABLE VIII

THE AVERAGE RESULTS FOR THE *scene* BENCHMARK SET. BEST RESULTS FOR EACH EVALUATION MEASURE ARE BOLDED.

	BP-MLL	CART-M	TCART-M*g*	TCART-M*i*	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-k NN	RAkEL	ECC	RFML-C4.5	RF-PCT
Hamming Loss	0.250	0.047	0.047	0.046	**0.045**	0.064	0.048	0.048	0.051	0.053	0.058	0.051	**0.045**	0.049	0.047	0.046
Accuracy	0.202	0.430	0.459	0.459	0.446	0.334	0.459	0.388	**0.478**	0.418	0.196	0.319	0.428	0.462	0.374	0.416
Precision	0.214	0.680	0.678	0.679	0.703	0.464	0.650	0.624	0.616	0.623	0.415	0.587	0.708	0.652	0.690	**0.709**
Recall	**0.849**	0.484	0.535	0.539	0.497	0.507	0.557	0.453	0.610	0.487	0.229	0.358	0.469	0.560	0.398	0.452
Subset Accuracy	0.002	0.133	0.134	0.137	**0.149**	0.000	0.117	0.097	0.145	0.140	0.002	0.062	0.136	0.131	0.124	0.131
F1 score	0.315	0.536	0.571	0.572	0.582	0.484	0.600	0.525	**0.613**	0.546	0.295	0.445	0.564	0.602	0.505	0.552
Micro-precision	0.187	0.713	0.680	0.694	0.721	0.492	0.652	0.687	0.597	0.613	0.601	0.684	0.743	0.642	**0.768**	0.738
Macro-precision	0.255	**0.294**	0.277	0.289	0.258	0.260	0.205	0.242	0.241	0.142	0.023	0.170	0.222	0.249	0.245	0.233
Micro-recall	**0.821**	0.450	0.506	0.510	0.464	0.472	0.532	0.438	0.585	0.440	0.246	0.353	0.435	0.532	0.366	0.422
Macro-recall	**0.505**	0.267	0.274	0.272	0.120	0.146	0.139	0.120	0.163	0.107	0.030	0.075	0.097	0.129	0.082	0.100
Micro-F1	0.302	0.551	0.580	0.578	0.564	0.482	0.585	0.535	**0.591**	0.512	0.349	0.466	0.548	0.582	0.496	0.537
Macro-F1	**0.274**	0.273	0.272	**0.274**	0.143	0.153	0.149	0.143	0.167	0.115	0.026	0.087	0.115	0.140	0.102	0.122
Ranking Loss	0.165	0.109	0.091	0.089	0.084	0.083	**0.078**	0.177	0.183	0.120	0.114	0.093	0.283	0.238	0.083	0.079
OneError	0.793	0.235	0.225	0.227	0.237	0.238	0.231	0.269	0.314	0.309	0.392	0.280	0.290	0.247	**0.219**	0.221
Coverage	16.845	15.824	13.684	13.248	12.530	12.437	**11.763**	22.746	24.190	17.010	14.920	13.181	30.509	27.760	12.485	12.074
Average Precision	0.335	0.667	0.685	0.686	0.693	0.695	**0.699**	0.604	0.604	0.629	0.546	0.635	0.522	0.576	0.680	0.698

TABLE IX

THE AVERAGE RESULTS FOR THE *enron* BENCHMARK SET. BEST RESULTS FOR EACH EVALUATION MEASURE ARE BOLDED.

	BP-MLL	CART-M	TCART-M*g*	TCART-M*i*	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-k NN	RAkEL	ECC	RFML-C4.5	RF-PCT
Hamming Loss	0.651	0.020	0.020	0.022	0.077	0.077	0.017	**0.012**	**0.012**	0.013	0.023	0.016	**0.012**	0.014	0.022	0.014
Accuracy	0.029	0.340	0.354	0.374	0.206	0.211	0.656	0.658	0.713	**0.730**	0.228	0.528	0.673	0.611	0.250	0.591
Precision	0.029	0.382	0.403	0.340	0.211	0.217	0.695	0.697	0.762	**0.797**	0.285	0.575	0.730	0.662	0.284	0.635
Recall	**0.829**	0.345	0.360	0.351	0.735	0.754	0.795	0.801	0.760	0.740	0.227	0.547	0.679	0.642	0.251	0.599
Subset Accuracy	0.000	0.293	0.300	0.302	0.000	0.000	0.486	0.480	0.610	**0.646**	0.177	0.462	0.607	0.526	0.216	0.538
F1 score	0.056	0.356	0.372	0.362	0.328	0.337	0.742	0.745	0.761	**0.768**	0.253	0.560	0.704	0.652	0.267	0.616
Micro-precision	0.031	0.841	0.833	0.823	0.225	0.229	0.669	0.667	0.807	0.796	0.826	0.807	0.881	0.834	0.884	**0.885**
Macro-precision	0.226	**0.519**	0.515	**0.521**	0.399	0.391	0.288	0.285	0.287	0.263	0.018	0.267	0.269	0.266	0.190	0.269
Micro-recall	**0.825**	0.353	0.359	0.384	0.725	0.739	0.782	0.787	0.742	0.720	0.227	0.522	0.600	0.624	0.237	0.569
Macro-recall	**0.622**	0.516	0.509	0.519	0.423	0.428	0.307	0.324	0.282	0.249	0.022	0.163	0.183	0.179	0.040	0.176
Micro-F1	0.058	0.491	0.495	0.521	0.343	0.350	0.721	0.722	**0.773**	0.756	0.356	0.634	0.714	0.714	0.374	0.693
Macro-F1	0.244	**0.516**	0.510	0.515	0.361	0.371	0.281	0.286	0.282	0.250	0.020	0.192	0.210	0.203	0.058	0.207
Ranking Loss	0.451	0.182	0.153	0.146	0.021	**0.019**	0.028	0.027	0.090	0.048	0.104	0.045	0.159	0.152	0.028	0.024
OneError	0.963	0.521	0.422	0.455	0.135	**0.123**	0.168	0.165	0.216	0.198	0.612	0.279	0.312	0.315	0.243	0.174
Coverage	21.143	9.404	8.247	7.814	1.610	**1.471**	2.036	1.832	5.324	3.033	5.813	2.844	8.520	7.994	1.889	1.619
Average Precision	0.106	0.520	0.609	0.590	0.896	**0.901**	0.864	0.862	0.786	0.823	0.522	0.784	0.676	0.684	0.817	0.868

TABLE X

THE AVERAGE RESULTS FOR THE *medical* BENCHMARK SET. BEST RESULTS FOR EACH EVALUATION MEASURE ARE BOLDED.

rithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[17] S.-H. Park and J. Fürnkranz, "Efficient pairwise classification," in *European Conference on Machine Learning*. Springer, 2007, pp. 658–665.

[18] E. L. Mencía, S.-H. Park, and J. Fürnkranz, "Efficient voting prediction for pairwise multilabel classification," *Neurocomputing*, vol. 73, no. 7, pp. 1164–1176, 2010.

[19] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Effective and efficient multilabel classification in domains with large number of labels," in *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08)*, 2008, pp. 30–44.

[20] A. Clare and R. D. King, "Knowledge discovery in multi-label phenotype data," in *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, 2001, pp. 42–53.

[21] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An ensemble method for multilabel classification," in *European Conference on Machine Learning*. Springer, 2007, pp. 406–417.

[22] F. Comite, R. Gilleron, and M. Tommasi, "Learning multi-label alternating decision tree from texts and data," in *Lecture Notes in Computer Science*, vol. 2734. Springer-Verlag, 2003, pp. 35–49.

[23] WHO. International classification of diseases. [Online]. Available: http://www.who.int/classifications/icd/en/