An approximated decision-theoretic algorithm for minimization of the Tversky loss under the multi-label framework

Paweł Trajdos¹ · Marek Kurzynski¹

Received: 8 February 2017 / Accepted: 1 August 2017 / Published online: 13 September 2017
© The Author(s) 2017. This article is an open access publication

Abstract In this paper, we addressed the problem of building a decision-theoretic classifier tailored for minimizing the Tversky loss under the framework of multi-label classification. The proposed approach is a generalization of the Dembczyński F_β measure optimization algorithm. The introduced technique is based on a series of discrete linear approximations of the Tversky measure. The approximated criterion is then optimized using original optimization algorithm. To assess quality of classification results produced by the designed strategy and compare its outcome with the results obtained by the state-of-the-art approaches, we conducted an experimental study on 24 benchmark datasets. The investigated methods were compared with respect to eleven different quality criteria. We considered quality criteria belonging to three main groups, i.e., example-based, micro-averaged and macro-averaged. During the experimental study, we considered four testing scenarios. Two of them deal with a symmetric variant of the Tversky loss. Remaining scenarios examine asymmetric Tversky loss. The study shows that, in general, the proposed method is comparable to the Dembczyński approach. However, for both symmetric scenarios and one asymmetric scenario, the average ranks suggest that the proposed approach achieves better classification quality in terms of the example-based Tversky measure. This is an important result because the proposed method was designed to optimize the above-mentioned quality indicator. Additionally, the introduced procedure can outperform the

reference methods with respect to the zero-one loss under all testing scenarios.

Keywords Multi-label classification · Tversky index · F measure · Plug-in-rule classifier · structured output

1 Introduction

Traditional single-label classification is concerned with learning from a set of examples which are assigned to a label (class) from a disjoint set of labels. In other words, only one label is relevant to the given object. Nevertheless, the assumption that labels are disjoint do not always hold. This issue emerges in many real-life recognition tasks: for example, a photograph may be tagged with such labels as lake, forest, sky and mountains. This label set constitutes a complete description of the object, and omitting one of labels in the classifier outcome must be considered as a classification error. Consequently, traditional single-label classification methods cannot directly be employed to solve a problem which violates the assumption that classes are disjoint. Strictly speaking, they are capable of predicting only a single category per object which in case of multi-label data is insufficient. A solution to this problem is to employ the multi-label (ML) classification framework which can be seen as a generalization of the classical recognition task [20, 49]. In the multi-label recognition, it is assumed that an object is simultaneously assigned to more than one class. What is more, the multi-label learning also considers two extraordinary cases: an object belongs to all possible labels or to none of the labels (interpretation of these cases is specific to the domain of the considered task).

Multi-label learning is employed in a variety of practical applications but the most widespread ones are: text

✉ Paweł Trajdos
pawel.trajdos@pwr.wroc.pl

¹ Department of Systems and Computer Networks, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland

classification [29, 30] and multimedia classification including classification of video objects [12], images [4, 57] and music [43]. Another important field of application is bioinformatics where multi-label classification is a powerful tool for prediction of: gene functions [44], protein functions [55, 56] or drug resistance [24], to name only a few. Nowadays, multi-label classification is becoming more and more common among the machine learning society. This growth is mainly caused by an increasing amount of data which can be effectively modelled using the multi-label framework [19]. A great example of this phenomenon is the growth in number of protein and nucleotide sequences stored in EMBL databases [58].

This paper is aimed at providing a flexible, effective and efficient classification procedure that is tailored to optimize the *Tversky measure*. We are focused on the Tversky measure because it is a far more general quality indicator than commonly used loss functions such as F_β measure, Jaccard measure, the zero-one loss, false discovery rate or false negative rate. Namely, all aforesaid quality indicators can be expressed in terms of the Tversky loss by setting proper values of its parameters. Consequently, building an effective classifier aimed at optimization of this measure can provide a general tool that can cover many user-specific loss criteria. To achieve our goal, we propose a generalization of the method described by Dembczyński in [8]. The introduced technique approximates the Tversky measure using a set of discrete linear functions. The task described by the linear approximation is solved using the inner–outer minimization approach in a way analogous to the solution proposed in the original approach.

The proposed method was also experimentally compared to the reference methods. The experimental procedure employs 24 benchmarks datasets and 11 quality criteria. We considered quality criteria belonging to three main groups, i.e., example-based, micro-averaged and macro-averaged. During the experimental study, we considered four testing scenarios. Two of them deal with a symmetric variant of the Tversky loss. Remaining scenarios examine asymmetric Tversky loss. The conducted experimental study provides an empirical evidence that the proposed method can outperform reference algorithms in terms of example-based Tversky measure and the zero-one loss.

The paper is organized as follows. Section 2 provides a description of the work related to the topic of this paper. Section 3 introduces a formal description of the proposed method. The next Sect. 4 describes experimental setup. The obtained results are presented and discussed in Sect. 5. The paper is concluded in Sect. 6.

2 Related work

Multi-label classifiers predict a vector response. Due to the complexity of the output structure of multi-label models, it is possible to evaluate the quality of multi-label classification using many different criteria such as *Hamming distance*, *zero-one subset loss* [10] or F_β measure [41]. The criteria also differ on the method of combining well-known single-label quality measures to produce a multi-label quality criterion. Namely, we can distinguish three possible ways: *example-based*, *macro-averaged* and *micro-averaged* [33]. Algorithms are usually designed to optimize a chosen quality measure, and the classifier designed to optimize one quality criterion is usually suboptimal under another quality criterion [10]. During this study, our focus is put on algorithms which are tailored to maximize classification quality expressed in terms of an asymmetric information retrieval measure known as the *Tversky measure* [53]. The Tversky measure is more general than F_β measure, so building an effective algorithm dedicated to this function allows us to express and optimize a wider range of quality criteria such as F_β measure, *Jaccard measure* [26] or zero-one subset loss [10] (relations between aforesaid measures are discussed in Sect. 3.1). Multi-label approaches aimed at dealing with optimization of the aforesaid quality criteria (including the Tversky measure) can be basically divided into empirical utility maximization methods and decision-theoretic methods [35].

The empirical utility maximization approaches build classifiers which are designed to obtain the optimal value of quality measure defined in the learning set. Learning these models is usually done by determining values of the model parameters that optimize the quality criterion. After that, the model with determined parameters is used to calculate classifier output for a test instance. Algorithms from this group are commonly based on structured SVMs [14, 37], thresholding strategies [38–40] or regression [27]. The structured output SVM is a generalization of classical SVM algorithm [48]. The procedure is tailored to deal with the classification problems whose output is more complex than single class. The approach can be adopted to the task of multi-label classification in a straightforward way [14]. As in the classical SVM approach, it is also possible to utilize different kernel functions under the considered approach [60]. The thresholding procedures mainly employ a state-of-the-art multi-label classifier that returns a set of label supports. The outcome of the classifier is then converted to binary prediction using a set of dynamically determined thresholds [39, 40]. The aforementioned approaches were originally harnessed to optimize the F_β measure, but they can also be employed to optimize the Tversky measure.

On the other hand, the decision-theoretic methods use the learning set to estimate parameters of the underlying

probability model. In the inference phase, the values of the probability distributions are calculated according to the estimated model. There were proposed a few methods based on this framework [6, 8, 28, 35]. Chai [6] tackled the posed problem by expressing the expected loss as a recursive function, and then, he solved the arisen optimization task using dynamic programming. Another approach to tackle with the above-mentioned issue was provided by Jansche [28] who proved that the posed problem can be effectively solved via inner and outer optimization. To perform the optimization task, the space of all possible solutions is divided into non-overlapping equivalence classes. Then, the optimal solution is found for each of equivalence classes separately. Finally, the outer optimization is performed in order to determine the globally optimal solution. The author designed a method based on Lebesgue integral and two-tape automaton. Inspired by this methodology, Dembczyński et al. [8] proposed an alternative inner–outer optimization scheme which, in contrast to the formerly mentioned methods, does not make any assumptions about the underlying probability distribution. Unfortunately, the Dembczyński method, contrary to the remaining algorithms, cannot be directly employed to minimize the loss function based on the Tversky measure. An alternative set of equivalence classes was described by Nan et al. [35]. Additionally, the authors presented a heuristic procedure that allows them to reduce the computational burden.

Cheng et al. analysed the Classifier Chain approach [40], which extends the basic *binary relevance* approach [1], under probabilistic formalism. Their work showed that the original method is a simplified strategy of a more general framework [7] of conditional joint mode estimation. During the inference phase, the simplified approach performs greedy search procedure that follows only a single path in a tree of all possible solutions. The authors proposed to employ an inference algorithm that performs exhaustive search in order to determine the optimal solution. Although the routine enables us to find the optimal solution in terms of any loss function (including F_β and Tversky), its computational complexity grows exponentially (2^L , where L is the number of labels) with the number of labels. As a consequence, the computational burden of the approach is extremely high and it can be directly employed only when the number of labels is relatively low. However, the above-mentioned drawback was dealt with in an application of heuristic methods of finding the optimal path in the tree of possible solutions [10, 32].

3 Proposed method

3.1 Preliminaries

In the introductory section, we outlined the basic description of the multi-label classification task. Now, let us define a

more formal description of the investigated issue. An object x is now interpreted as a vector $x = [x_1, x_2, \dots, x_d]$ that comes from the d -dimensional input space $\mathcal{X}$. The set labels related to the object is indicated by a binary vector of length L : $y = [y_1, y_2, \dots, y_L]$ and $y_i = 1$ ($y_i = 0$) denotes that i -th label is relevant (irrelevant) to the object x . As a consequence, the output space is defined as $\mathcal{Y} = \{0, 1\}^L$ which denotes a set of all possible binary vectors of length L . Additionally, it is assumed that object x and its set of labels y are realizations of corresponding random vectors $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_d]$, $\mathbf{Y} = [\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_L]$ and the joint probability distribution $P(\mathbf{X}, \mathbf{Y})$ on $\mathcal{X} \times \mathcal{Y}$ is known.

Relevant labels are assigned to instances by an unknown mapping $f : \mathcal{X} \mapsto \mathcal{Y}$. A classifier function $\psi : \mathcal{X} \mapsto \mathcal{Y}$ is an approximation of the unknown mapping. Finding the classifier function is usually stated as a problem of optimal decision making given loss function. The loss function $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \mapsto \mathcal{R}_+$ assesses similarity between vectors from the output space. Without loss of generality, it is assumed that only normalized loss functions $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \mapsto [0, 1]$ are considered. In general, the optimal decision making aims to find a classifier ψ^* that minimizes the expected loss over the joint probability distribution $P(\mathbf{X}, \mathbf{Y})$:

$$\psi^* = \operatorname{argmin}_{\psi} \mathbb{E}_{\mathbf{X}\mathbf{Y}}[\mathcal{L}(\psi(\mathbf{X}), \mathbf{Y})], \quad (1)$$

where $\mathbb{E}$ is the expected value operator. The above-mentioned classifier can be found in a pointwise way by the Bayes optimal decisions

$$\begin{aligned} h^*(x) &= \operatorname{argmin}_{h \in \mathcal{Y}} \mathbb{E}_{\mathbf{Y}|\mathbf{X}=x}[\mathcal{L}(h, \mathbf{Y})] \\ &= \operatorname{argmin}_{h \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} \mathcal{L}(h, y) P(y|x) \end{aligned} \quad (2)$$

where $h^*(x) = [h_1^*(x), h_2^*(x), \dots, h_L^*(x)] \in \mathcal{Y}$ denotes an optimal prediction for instance x and $P(y|x) = P(Y = y|X = x)$ is the conditional probability distribution of vector y given an object x . It is clear that the optimal solution cannot be efficiently found via exhaustive search because the size of the output space is $|\mathcal{Y}| = 2^L$.

Although there is a bunch of loss functions that can be adopted under the multi-label classification methodology, in this paper we are focused on the learning algorithms that optimize the Tversky loss $T_{\gamma, \delta}$, which is defined as follows: above-mentioned loss function

$$T_{\gamma, \delta}(h(x), y(x)) = 1 - \frac{h(x) \cdot y(x)}{\gamma \|h(x)\|_1 + \delta \|y(x)\|_1 + \eta h(x) \cdot y(x)}, \quad (3)$$

where $h(x) \in \mathcal{Y}$ is the prediction of a classifier, $y(x) \in \mathcal{Y}$ indicates the ground truth labels for the instance x , $\eta = 1 - \gamma - \delta$, $\gamma > 0$ and $\delta > 0$ can be interpreted as weights related to *False Positive rate* and *False Negative rate*,

respectively. A growth in one of those weights increases the penalty related to the type of error associated with the weight. Additionally $\|\cdot\|_1$ is the L_1 norm, and $h(x) \cdot y(x)$ is a dot product of a given pair of vectors. For a special case when $\|h(x)\|_1 = \|y(x)\|_1 = 0$, it is assumed that $T_{\gamma,\delta}(h(x), y(x)) = 0$. The above-mentioned loss function is worth considering because it is more general than the loss functions that are usually applied to build a multi-label classifier. Namely, using this loss function, it is possible to express such loss functions as $F_\beta(h(x), y(x))$, the Jaccard loss $J(h(x), y(x))$ or zero-one loss $Z(h(x), y(x))$:

$$F_1(h(x), y(x)) = T_{\gamma=0.5, \delta=0.5}(h(x), y(x))$$

$$F_\beta(h(x), y(x)) = T_{\gamma=\frac{1}{1+\beta^2}, \delta=\frac{\beta^2}{1+\beta^2}}(h(x), y(x))$$

$$J(h(x), y(x)) = T_{\gamma=1, \delta=1}(h(x), y(x))$$

$$Z(h(x), y(x)) = T_{\gamma \rightarrow \infty, \delta \rightarrow \infty}(h(x), y(x))$$

It is also possible to express such measures as false discovery rate $FDR(h(x), y(x))$ and false negative rate $FNR(h(x), y(x))$:

$$FDR(h(x), y(x)) = T_{\gamma=1, \delta=0}(h(x), y(x))$$

$$FNR(h(x), y(x)) = T_{\gamma=0, \delta=1}(h(x), y(x))$$

The differences between aforementioned loss functions become clearer when we express them using binary confusion matrix resulting from comparison of two binary strings $h(x)$ and $y(x)$ (Table 1). Entries of the matrix are defined as follows:

$$TP = \sum_{i=1}^L h_i(x) y_i(x) \quad (4)$$

$$TN = \sum_{i=1}^L [1 - h_i(x)] [1 - y_i(x)] \quad (5)$$

$$FP = \sum_{i=1}^L h_i(x) [1 - y_i(x)] \quad (6)$$

$$FN = \sum_{i=1}^L [1 - h_i(x)] y_i(x). \quad (7)$$

Table 1 Confusion matrix resulting from comparison of two binary strings $h(x)$ and $y(x)$

		$y(x)$	
		1	0
$h(x)$	1	TP	FP
	0	FN	TN

Then, the measures can be rewritten:

$$Z(h(x), y(x)) = \llbracket FN + FP = 1 \rrbracket \quad (8)$$

$$F_\beta(h(x), y(x)) = \frac{\beta^2 FN + FP}{(1 + \beta^2) TP + \beta^2 FN + FP} \quad (9)$$

$$J(h(x), y(x)) = \frac{FN + FP}{TP + FN + FP} \quad (10)$$

$$T_{\gamma,\delta}(h(x), y(x)) = \frac{\delta FN + \gamma FP}{TP + \delta FN + \gamma FP}, \quad (11)$$

where $\llbracket \cdot \rrbracket$ is the Iverson bracket.

3.2 Bayes classifier for the F_β loss

In this section, we describe the original method proposed by Dembczyński [8]. The method is based upon the inner–outer framework which was proven to be an efficient way to find the optimal classifier tailored for the F_β loss [28]:

$$F_\beta(h(x), y(x)) = 1 - \frac{(1 + \beta^2) h(x) \cdot y(x)}{\beta^2 \|y(x)\|_1 + \|h(x)\|_1}. \quad (12)$$

In the case of the F_β loss, the Bayes classifier (2) is given by

$$h^*(x) = \operatorname{argmin}_{h \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} \left(1 - \frac{(1 + \beta^2) h \cdot y}{\beta^2 \|y\|_1 + \|h\|_1} \right) P(y|x) \quad (13)$$

Next, the posed problem (13) is solved via inner and outer maximizations. In order to perform the inner maximization, the space of all possible solutions $\mathcal{Y}$ is partitioned into $L + 1$ non-overlapping equivalence classes. Each equivalence class contains binary vectors $h \in \mathcal{Y}$ with the number of ones equal to $K = \|h\|_1$. As a consequence, we denote the equivalence class as a set $\mathcal{H}_K = \{h : h \in \mathcal{Y}, \|h\|_1 = K\}$ and $K \in \{0, 1, \dots, L\}$. Analogously, the partitioning into the equivalence can also be employed to vector y from the Eq. (13). The sum of ones in this vector is denoted by $S = \|y\|_1$. Then, the optimization problem can be solved for each of equivalence classes separately. Before we define inner optimization problem, let us introduce a subset of $\mathcal{Y}$ in which the number of ones in vectors sums up to s :

$$S_s = \{y \in \mathcal{Y}, y : \|y\|_1 = s\}. \quad (14)$$

Then, for each equivalence class $\mathcal{H}_K$ an optimal prediction is described by:

$$h^*(x)_K = \operatorname{argmin}_{h \in \mathcal{H}_K} \sum_{S=0}^L \sum_{y \in S_S} \left(P(y|x) - \frac{(1 + \beta^2) P(y|x)}{\beta^2 S + K} \sum_{i=1}^L h_i y_i \right) \quad (15)$$

$$= \operatorname{argmax}_{h \in \mathcal{H}_K} \sum_{S=1}^L \frac{(1 + \beta^2)}{\beta^2 S + K} \sum_{y \in S_S} P(y|x) \sum_{i=1}^L h_i y_i \quad (16)$$

After that, by swapping sums in (16) and skipping term $(1 + \beta^2)$, the authors obtain

$$\begin{aligned} h^*(x)_K &= \arg \max_{h \in H_K} \sum_{i=1}^L h_i \sum_{S=1}^L \frac{P(y_i = 1, \|y\|_1 = S|x)}{\beta^2 S + K} \\ &= \arg \max_{h \in H_K} \sum_{i=1}^L h_i \sum_{S=1}^L \frac{P(y_i = 1|x)P(\|y\|_1 = S|y_i = 1, x)}{\beta^2 S + K} \\ &= \arg \max_{h \in H_K} \sum_{i=1}^L h_i \Delta(i, K). \end{aligned} \quad (17)$$

where

$$P(y_i = 1|x) = \sum_{y \in \mathcal{Y}: y_i=1} P(y|x) \quad (18)$$

is the probability that i -th bit is set given x and

$$P(\|y\|_1 = S|y_i = 1, x) = \sum_{y \in S_S} P(y|x, y_i = 1) \quad (19)$$

is the probability that the number of ones in vector y is S given $y_i = 1$ and x . Since we must set $h_i = 1$ for only K positions, the optimization problem can be solved optimally by setting $h_i = 1$ for the top K values of $\Delta(i, K)$.

To perform the outer optimization, it is also necessary to calculate the expected loss related to the previously determined solution $\mathbb{E}_{Y|X=x}[F_\beta(h^*(x)_K, Y)]$:

$$\mathbb{E}_{Y|X=x}[F_\beta(h^*(x)_K, Y)] = 1 - (1 + \beta^2) \sum_{i=1}^L h_i^*(x)_K \Delta(i, K). \quad (20)$$

Additionally, the expected value associated with inner optimization task for $K = 0$ is:

$$\begin{aligned} \mathbb{E}_{Y|X=x}[F_\beta(h^*(x)_0, Y)] &= \sum_{\substack{y \in \mathcal{Y} \\ y: \|y\|_1 = 0}} P(y|x) \\ &= P(\|y\|_1 = 0|x) \end{aligned} \quad (21)$$

Finally, the outer minimization finds the best prediction among the predictions produced for each equivalence class $\mathcal{H}_K$:

$$h^*(x) = \arg \min_{h \in \{h^*(x)_0, h^*(x)_1, \dots, h^*(x)_L\}} \mathbb{E}_{Y|X=x}[F_\beta(h, Y)] \quad (22)$$

and its solution is found by checking all possible $L + 1$ ($K \in \{0, 1, 2, \dots, L\}$) expected values.

3.3 Proposed method

In this subsection, an approximated method of building decision-theoretic classifier for the Tversky measure is introduced. We begin with considering the Bayes classifier for the $T_{\gamma, \delta}$:

$$\bar{h}^*(x) = \arg \min_{h \in \mathcal{Y}} \sum_{y \in \mathcal{Y}} \left(1 - \frac{h \cdot y}{\delta \|y\|_1 + \gamma \|h\|_1 + \eta h \cdot y} \right) P(y|x) \quad (23)$$

After transformation analogous to (15), we obtain:

$$\begin{aligned} \bar{h}^*(x)_K &= \arg \min_{h \in H_K} \sum_{S=0}^L \sum_{y \in S_S} \left(P(y|x) - \frac{P(y|x) \sum_{i=1}^L h_i y_i}{\delta S + \gamma K + \eta h \cdot y} \right) \\ &= \arg \max_{h \in H_K} \sum_{S=1}^L \sum_{y \in S_S} \frac{P(y|x) \sum_{i=1}^L h_i y_i}{\delta S + \gamma K + \eta h \cdot y}. \end{aligned} \quad (24)$$

Although a transformation analogous to (17) cannot eliminate term $\eta h \cdot y$ from the denominator of (24), it is impossible to perform directly the optimization procedure described in the previous subsection. In order to apply the aforementioned procedure, one can simply remove the term but this simplification can lead to a coarse approximation. In this paper, we propose more accurate approximation which does not induce a significant increase in the computational complexity.

We start with a simple observation that for given K , S and x , each term of expected loss function (15) is a linear discrete function of $h \cdot y$. This remark leads to a conclusion that the approximation of the Tversky expected loss can be efficiently computed using linear approximations:

$$\begin{aligned} g(K, S, x, h \cdot y = z) &= d(K, S) \sum_{y \in S_S} P(y|x) \\ &\quad - c(K, S) \sum_{y \in S_S} P(y|x) z. \end{aligned} \quad (25)$$

For the sake of simplicity, let us introduce the following notation:

$$l(K, S, x, h \cdot y = z) = \sum_{y \in S_S} P(y|x) - \sum_{y \in S_S} \frac{P(y|x) z}{\delta S + \gamma K + \eta z}, \quad (26)$$

$$\delta S + \gamma K = R. \quad (27)$$

Now, please note that for fixed L , K and S , the term $h \cdot y$ is bounded as follows:

$$h \cdot y \leq \min(K, S) = b_u(K, S) \quad (28)$$

$$h \cdot y \geq \max(0, K + S - L) = b_l(K, S). \quad (29)$$

In the interval defined by $b_l(K, S)$ and $b_u(K, S)$, the original loss function can be approximated using a linear function (25) whose parameters $c(K, S)$ and $d(K, S)$ are calculated from the following system of linear equations

$$\begin{cases} g(K, S, x, b_u(K, S)) = l(K, S, x, b_u(K, S)), \\ g(K, S, x, b_l(K, S)) = l(K, S, x, b_l(K, S)) \end{cases} \quad (30)$$

which gives (details are shown in "Appendix 2.1")

$$\begin{cases} c(K, S) = \frac{R}{(R + \eta b_u(K, S))(S + \eta b_l(K, S))}, \\ d(K, S) = 1 - \frac{\eta b_u(K, S) b_l(K, S)}{(R + \eta b_u(K, S))(R + \eta b_l(K, S))}. \end{cases} \quad (31)$$

Example 1 Let us show an example of approximation $T_{\gamma=10, \delta=1}$ loss using linear functions. Throughout this example, we assumed that $L = 20, K=15$ and $S = 15$. For this case, the bounds of $h \cdot y$ are $b_u(15, 15) = 15$ and $b_l(15, 15) = 10$. Considering all the above, we construct a linear approximation of the $T_{\gamma=10, \delta=1}$ in the interval $[10; 15]$. This approximation is presented in Fig. 1. As we can see, under such circumstances, the difference between our approximation and approximation performed using Dembczyński approach is substantial (red and green lines).

Finally, we define an approximated inner classifier:

$$\begin{aligned} \tilde{h}^*(x)_K = \arg \min_{h \in H_K} \sum_{S=0}^L \sum_{y \in S_S} & \\ \left(d(K, S) - c(K, S) \sum_{i=1}^L h_i y_i \right) P(y|x) & \end{aligned} \quad (32)$$

$$\begin{aligned} = \arg \min_{h \in H_K} \left\{ \sum_{S=0}^L \sum_{y \in S_S} P(y|x) d(K, S) \right. & \\ \left. - \sum_{S=1}^L \sum_{y \in S_S} P(y|x) c(K, S) \sum_{i=1}^L h_i y_i \right\}. & \end{aligned} \quad (33)$$

And since the term $d(K, S)$ does not depend on h but only on K , which is constant in each inner optimization problem:

$$\tilde{h}^*(x)_K = \arg \max_{h \in H_K} \sum_{S=0}^L \sum_{y \in S_S} P(y|x) c(K, S) \sum_{i=1}^L h_i y_i. \quad (34)$$

Fig. 1 An example of the Tversky ($T_{\gamma=10, \delta=1}$) measure approximation. The measure is expressed as a function of the number of true positives ($h \cdot y$) while the number of labels L , the number of relevant labels in h and y are fixed to K and S respectively. The approximation interval computed according to (28) and (29) is $[10; 15]$

that can be efficiently found via the inner optimization approach proposed by Dembczyński:

$$\begin{aligned} \tilde{h}^*(x)_K &= \arg \max_{h \in H_K} \sum_{i=1}^L h_i \sum_{S=1}^L \\ & c(K, S) P(y_i = 1|x) P(\|y\|_1 = S | y_i = 1, x) \\ &= \arg \max_{h \in H_K} \sum_{i=1}^L h_i \tilde{\Delta}(i, K). \end{aligned} \quad (35)$$

The expected loss related to the determined classifier is:

$$\begin{aligned} \mathbb{E}_{Y|X=x} [T_{\gamma, \delta}(\tilde{h}^*(x)_K, Y)] &= \sum_{S=0}^L P(\|y\|_1 = S | x) d(K, S) \\ & - \sum_{i=1}^L \tilde{h}_i^*(x)_K \tilde{\Delta}(i, K), \end{aligned} \quad (36)$$

where $\tilde{\Delta}(i, K)$ is defined in a way analogous to the transformation applied in (17). As a consequence, the outer optimization algorithm does not differ from the procedure described in the previous subsection. That is, the probability of getting zero vector is:

$$\mathbb{E}_{Y|X=x} [T_{\gamma, \delta}(\tilde{h}^*(x)_0, Y)] = P(\|y\|_1 = 0 | x), \quad (37)$$

and outer optimization is performed according to:

$$\tilde{h}^*(x) = \arg \min_{h \in \{\tilde{h}^*(x)_0, \tilde{h}^*(x)_1, \dots, \tilde{h}^*(x)_L\}} \mathbb{E}_{Y|X=x} [T_{\gamma, \delta}(h, Y)] \quad (38)$$

3.4 Plug-in rule classifier for the $T_{\gamma, \delta}$ loss

The previous section provides us with description of the Bayes classifier tailored for the $T_{\gamma, \delta}$ loss function. The description assumes that all considered probability distributions are known, but in real-life classification tasks this assumption does not hold. In such situations, an approach referred as *plug-in rule* classifier [31] can be employed. The plug-in rule approach consists in estimating the unknown probabilities that are calculated on the basis of a training set

$$\mathcal{D} = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\} \quad (39)$$

and then plugged into the formula of the Bayes classifier.

The above-defined Bayes classifier requires $L^2 + 2L$ probabilities to be estimated, namely:

- L^2 for $P(\|y\|_1 = S | y_i = 1, x)$;
- L for $P(y_i = 1 | x)$;
- and L for $P(\|y\|_1 = S | x)$.

Calculation of these values can be efficiently done by employing a set of $2L + 1$ multinomial regression models or classifiers (only classifiers that return an estimation of the posterior probability are considered).

Probabilities $P(\|y\|_1 = S | y_i = 1, x)$ are estimated using L models. First step in the procedure of building the models is to make sets $\mathcal{D}_i$ that contains only objects for which $y_i^{(j)} = 1$. After that, a group of estimators is learned on the sets transformed in the following way

$$(x^{(j)}, y^{(j)}) \in \mathcal{D}_i \mapsto (x^{(j)}, \|y\|_1) \quad \forall \quad j \in 1, 2, \dots, |\mathcal{D}|. \quad (40)$$

Probabilities $P(y_i = 1 | x)$ are modelled by applying the binary relevance transformation on the training set:

$$(x^{(j)}, y^{(j)}) \in \mathcal{D} \mapsto (x^{(j)}, y_i^{(j)}). \quad (41)$$

The transformation produces L one-vs-rest binary sets corresponding to each label.

Finally, we obtain an estimation of $P(\|y\|_1 = S | x)$ by performing a transformation

$$(x^{(j)}, y^{(j)}) \in \mathcal{D} \mapsto (x^{(j)}, \|y^{(j)}\|_1). \quad (42)$$

followed by learning of a related model.

3.5 System architecture

The description of learning and inference phases are provided in Figs. 2 and 3.

```

Input data:
   $\mathcal{D}$  - training set;
BEGIN
  For  $l \in \{1, 2, \dots, L\}$ :
    Build estimator of  $P(\|y\|_1 = S | y_i = l, x)$  using  $\mathcal{D}$ ;
    Build estimator of  $P(y_l = 1 | x)$  using  $\mathcal{D}$ ;
  End For;
  Build estimator of  $P(\|y\|_1 = S | x)$  using  $\mathcal{D}$ ;
END

```

Fig. 2 Pseudocode of the learning procedure

```

Input data:
   $x$  - the testing point;
BEGIN
  /*
  Perform the inner optimisation for
  each equivalence class  $\mathcal{H}_K$ .
  Employ plug-in probability estimations.
  */
  For  $K \in \{1, 2, \dots, L\}$ ;
    Find  $\tilde{h}^*(x)_K$  using (35);
    Calculate expected loss of  $\tilde{h}^*(x)_K$  using (36);
  End For;
  Calculate expected loss for zero vector using (37);
  Find final prediction  $\tilde{h}^*(x)$ 
  using outer optimisation (38);
  Return  $\tilde{h}^*(x)$ ;
END

```

Fig. 3 Classification procedure for given x

4 Experimental setup

During the experimental study, the proposed method was compared to five state-of-the-arts approaches. First of all, the most natural choice for a reference method is the one proposed by Dembczyński [8] upon which the developed approximation scheme is based. Additionally, we harnessed another decision-theoretic approach introduced by Jan-sche [28]. We also employed an algorithm that follows the empirical utility maximization framework, namely Pillai thresholding procedure [39]. We considered four versions of the aforesaid algorithms, and they differ according to values of parameters γ, δ of the Tversky loss. The following parameter values were examined:

- $\{\gamma = 1; \delta = 1\}$ – corresponding to the well-known Jaccard measure [26] (known also as multi-label accuracy [49]) which is a common measure of multi-label classification quality.
- $\{\gamma = 10; \delta = 10\}$ – corresponding to the optimization of zero-one loss measure. As stated previously, the Tversky measure ideally approximates the zero-one loss when

$\gamma \rightarrow \infty$ and $\delta \rightarrow \infty$. However, preliminary experiments showed that increasing γ and δ above 10 does not induce significant increase in classification quality measured by the zero-one loss.

- $\{\gamma = 1; \delta = 10\}$ and $\{\gamma = 10; \delta = 1\}$ which are asymmetric variations of the above-mentioned quality criteria.

We did not consider the instantiation of algorithms with parameters set to $\{\gamma = 0.5; \delta = 0.5\}$ because, as it is shown in Sect. 3.3, under such setup the proposed algorithm is equivalent to the Dembczyński approach and the performance of this method was examined extensively [8]. Nevertheless, the Dembczyński algorithm can only be applied to minimize F_β loss, so its parameters must be set in such a way that allow us to obtain the best approximation of the Tversky measure with given parameters. Strictly speaking, β parameter was set to $\beta = \sqrt{\frac{\delta}{\gamma}}$. All of the algorithms, were implemented using the Naïve Bayes classifier [23] as a base single-label model. We utilized Naïve Bayes implemented in WEKA framework [22]. The classifier parameters were set to its defaults.

In the next section, the aforementioned approaches are numbered as follows:

1. the proposed approach (Sects. 3.3 and 3.4),
2. the Dembczyński algorithm [8],
3. the Jansche approach [28],
4. Pillai thresholding procedure [39],
5. Structured SVM approach [39],
6. the binary relevance method [49].

The performances of the above-mentioned classifiers were assessed using such quality criteria as *false discovery rate* (FDR, $1 - \text{precision}$), *false negative rate* (FNR, $1 - \text{recall}$) and the Tversky measure with parameters relevant to investigated algorithms [33]. We employed example-based (denoted by Ex), macro-averaged (denoted by Ma) and micro-averaged (denoted by Mi) measures. The algorithms were also compared with respect to the Hamming loss [10] and zero-one subset loss function.

The experimental evaluation was conducted on 24 benchmark multi-label datasets related to seven different domains: text categorization – 7 datasets, image annotation – 6, bioinformatics – 4, audio samples recognition

Table 2 Summary statistics of benchmark datasets

No.	Set name	N	d	L	LC	LD	LU
1	Birds [5]	645	279	19	1.014	.053	133
2	CAL500 [52]	502	242	174	26.044	.150	502
3	Corel16k001 [2]	13766	653	153	2.859	.019	4803
4	Corel16k002 [2]	13761	664	164	2.882	.018	4868
5	Corel16k003 [2]	13760	654	154	2.829	.018	4812
6	Drosophila_m. [55]	2605	42	22	2.656	.121	63
7	Emotions [47]	593	78	6	1.869	.311	27
8	Enron [17]	1702	1054	53	3.378	.064	753
9	Flags [21]	194	50	7	3.392	.485	54
10	Solar_flare1 [34]	323	28	3	0.232	.077	8
11	Solar_flare2 [34]	1066	30	3	0.209	.070	7
12	Genbase [13]	662	1213	27	1.252	.046	32
13	LangLOG [40]	1460	1079	75	1.180	.016	304
14	Mediamill [45]	43907	221	101	4.376	.043	6555
15	Medical [36]	978	1494	45	1.245	.028	94
16	MimlImg [61]	2000	140	5	1.236	.247	20
17	Rcv1subset1 [59]	6000	1783	101	2.880	.029	1028
18	Rcv1subset2 [59]	6000	1767	101	2.634	.026	954
19	Rcv1subset3 [59]	6000	1797	101	2.614	.026	939
20	Saccharomyces_c. [55]	3509	47	27	2.275	.084	109
21	Scene [4]	2407	300	6	1.074	.179	15
22	Slashdot [40]	3782	1101	22	1.181	.051	156
23	Water-quality [15]	1060	30	14	5.073	.362	825
24	Yeast [16]	2417	117	14	4.237	.303	198

N is the number of instances, d is the dimensionality of input space, L denotes the number of labels. LC, LD, LU are label cardinality, label density and the number of distinct label combinations, respectively

Table 3 Summarized results for the algorithms tailored to optimize the $\mathcal{T}_{1,1}$ loss function

No.	1	2	3	4	5	6	1	2	3	4	5	6
	$T_{\gamma=1,\delta=1}$ (Ex)						Zero-one					
Rnk	2.833	3.000	4.125	3.417	3.083	4.542	1.750	3.812	4.521	3.417	3.229	4.271
Frd	.019537						.000010					
Wp	–	1.00	.064	1.00	1.00	.147	–	.002	.002	.078	1.00	.003
	FDR (Ex)						FNR (Ex)					
Rnk	2.667	3.500	4.333	2.750	3.375	4.375	4.417	2.646	2.833	4.375	3.083	3.646
Frd	.008602						.005149					
Wp	–	0.058	0.003	1.000	1.000	0.028	–	0.027	1.000	1.000	0.754	1.000
	FDR (Ma)						FNR (Ma)					
Rnk	2.250	3.375	5.167	2.792	3.500	3.917	4.646	2.938	1.688	5.042	3.292	3.396
Frd	.000011						<10 ⁻⁶					
Wp	–	0.002	0.000	1.000	1.000	0.010	–	0.004	0.000	1.000	1.000	0.159
	FDR (Mi)						FNR (Mi)					
Rnk	2.458	3.833	5.083	2.250	3.542	3.833	4.771	2.854	1.854	4.292	3.458	3.771
Frd	.000005						.000004					
Wp	–	0.001	0.000	1.000	1.000	0.427	–	0.003	0.000	1.000	0.451	1.000
	$T_{\gamma=1,\delta=1}$ (Ma)						$T_{\gamma=1,\delta=1}$ (Mi)					
Rnk	3.667	2.750	3.417	4.375	2.708	4.083	3.000	3.208	3.958	2.875	3.292	4.667
Frd	.019537						.019537					
Wp	–	1.000	1.000	1.000	0.721	1.000	–	1.000	1.000	1.000	1.000	0.171
	Hamming											
Rnk	2.375	3.708	5.417	2.292	3.292	3.917						
Frd	<10 ⁻⁶											
Wp	–	0.000	0.000	1.000	1.000	0.005						

Rnk stands for average rank over all datasets, Frd is the p value obtained using the Friedman test and Wp denotes the p value associated with the Wilcoxon test

– 3, video annotation – 1, astronomy – 2 and environmental science – 1. The datasets are summarized in Table 2. The first column (named ‘No.’) shows the number of the dataset. The number is further used to denote the set in the experimental section. The second column of the table holds names of datasets and reference to related papers. Next three columns contain the number of instances in dataset, input space dimensionality and the number of labels, respectively. Another three columns provide more detailed information about dataset properties that is, set cardinality, density and the number of unique label combinations [50]. To be more formal, the aforementioned properties are defined as follows:

$$LC(\mathcal{D}) = \frac{1}{N} \sum_{i=1}^N \sum_{j=1}^L y_j^{(i)}, \quad (43)$$

$$LD(\mathcal{D}) = \frac{LC(\mathcal{D})}{L}, \quad (44)$$

$$LU(\mathcal{D}) = \left| \{y : \exists (x^{(i)}, y^{(i)}) \in \mathcal{D}, y^{(i)} = y\} \right|. \quad (45)$$

During the dataset-preprocessing stage, we applied a few transformations on datasets. First and foremost, all nominal attributes, except binary attributes, were converted into a set of binary variables. For example, a nominal feature with three possible values is transformed into a set of three binary variables and each binary variable indicates the occurrence of the associated nominal value. This approach is one of the simplest methods to replace nominal variables with binary variables [46].

In our study, we employed datasets that follow multi-instance-multi-label (MIML) framework [61], [55]. In those datasets, each object consists of a bag of instances tagged with a set of labels. In order to tackle this data, we follow the suggestion made in [62], and we transformed the set to single-instance multi-label data. Namely, we build a between-bag distance matrix using the Hausdorff distance [42], and then, we constructed a set of new points in Euclidean space using multidimensional scaling [3].

We also harnessed multi-target regression sets (Solar_flare1, Solar_flare2 and Water-quality) which were converted into multi-label data using a simple thresholding procedure. To be more precise, when the value of output variable for a given object is greater than zero, the corresponding label is

Table 4 Summarized results for the algorithms tailored to optimize the $\mathcal{T}_{10,10}$ loss function

No.	1	2	3	4	5	6	1	2	3	4	5	6
	$T_{\gamma=10,\delta=10}$ (Ex)						Zero-one					
Rnk	2.083	3.583	4.167	3.167	3.333	4.667	1.708	3.833	4.396	3.333	3.271	4.458
Frd	.000268						.000005					
Wp	–	0.003	0.000	0.087	1.000	0.001	–	0.003	0.002	0.022	1.000	0.001
	FDR (Ex)						FNR (Ex)					
Rnk	2.708	3.542	4.167	2.875	3.375	4.333	4.792	2.333	2.708	4.125	3.792	3.250
Frd	.032331						.000155					
Wp	–	0.101	0.048	1.000	1.000	0.048	–	0.001	1.000	0.376	0.087	0.376
	FDR (Ma)						FNR (Ma)					
Rnk	2.417	3.375	4.958	2.875	3.458	3.917	5.125	2.583	1.854	4.708	3.417	3.312
Frd	.000271						$<10^{-6}$					
Wp	–	0.051	0.000	1.000	1.000	0.088	–	0.001	0.000	1.000	0.007	0.038
	FDR (Mi)						FNR (Mi)					
Rnk	2.667	3.792	5.000	2.333	3.542	3.667	5.083	2.583	1.771	4.125	4.083	3.354
Frd	.000090						$<10^{-6}$					
Wp	–	0.095	0.000	1.000	1.000	1.000	–	0.000	0.000	0.013	0.021	0.087
	$T_{\gamma=10,\delta=10}$ (Ma)						$T_{\gamma=10,\delta=10}$ (Mi)					
Rnk	4.083	2.833	3.583	3.542	2.958	4.000	4.000	2.750	3.583	2.708	3.667	4.292
Frd	.102598						.032331					
Wp	–	1.000	1.000	1.000	0.572	1.000	–	1.000	1.000	1.000	1.000	1.000
	Hamming											
Rnk	2.375	3.875	5.292	2.333	3.250	3.875						
Frd	$<10^{-6}$											
Wp	–	0.004	0.000	1.000	1.000	0.014						

Rnk stands for average rank over all datasets, Frd is the p value obtained using the Friedman test and Wp denotes the p value associated with the Wilcoxon test

set to be relevant to this object. In case of Solar_flare sets, this transformation results in producing a multi-label set characterized by low label cardinality and density, and such a set should be considered as a kind of mischievous case.

The dimensionality of the input space for Rcv1subsetX datasets must be reduced since it causes the experimental software to run out of memory. To remove unnecessary attributes, we followed a filtering procedure suggested previously in [51]. Namely, we applied a simple frequency filter with the minimal number of word occurrences set to 50. As a consequence, the number of attributes is reduced from about 47k to about 1.8k.

Finally, training and testing sets were extracted using tenfold cross-validation. However, due to a large number of instances (nearly 44k instances), the number of cross-validation folds was reduced to three for Mediamill dataset. Despite the reduced number of cross-validation folds, the number of instances is large enough to provide a stable estimation of the classification quality criteria.

Statistical significance of obtained results was assessed using the Friedman test [18] and the post hoc Nemenyi test [11]. The corrected critical difference for the Nemenyi post hoc test is $CD_{\alpha=0.05} = 1.59$. Additionally, we performed

the Wilcoxon signed-rank test [11, 54]. For all tests, the significance level was set to $\alpha = 0.05$. To control family-wise error rates of the Wilcoxon testing procedure, the Holm approach of p value correction was employed [25].

5 Results and discussion

The experimental section was partitioned into two main subsections which contain the results related to the symmetric and asymmetric form of the Tversky loss. In the first section, we considered the symmetric case of the Tversky loss function, i.e., the loss function was instantiated with parameters set to $\gamma = \delta = 1$ (Jaccard loss) and $\gamma = \delta = 10$ (zero-one loss). The summarized results related to this scenario are presented in Tables 3, 4. The other subsection, on the other hand, is aimed at evaluating the introduced approach under two asymmetric variants of the Tversky loss. One is tuned to put a greater penalty to the false negatives $\{\gamma = 10, \delta = 1\}$, whereas the other increases the cost of false positives $\{\gamma = 1, \delta = 10\}$. The outcome of the asymmetric experiment is shown in Tables 5, 6. Both sections share the common format of a result table. Namely, the header table

Table 5 Results for the algorithms tailored to optimize the $\mathcal{T}_{10,1}$ loss function

No.	1	2	3	4	5	6	1	2	3	4	5	6
	$T_{\gamma=10,\delta=1}$ (Ex)						Zero-one					
Rnk	2.417	2.792	3.875	3.750	3.333	4.833	1.812	3.271	4.000	4.042	3.312	4.562
Frd	.000496						.000027					
Wp	–	0.727	0.005	0.642	1.000	0.000	–	0.005	0.003	0.110	1.000	0.002
	FDR (Ex)						FNR (Ex)					
Rnk	3.250	2.417	3.792	3.708	3.292	4.542	4.375	3.500	3.167	4.542	3.167	2.250
Frd	.004323						.000496					
Wp	–	1.000	0.400	1.000	1.000	0.018	–	0.287	1.000	1.000	0.079	0.107
	FDR (Ma)						FNR (Ma)					
Rnk	2.583	2.917	4.208	2.792	3.500	5.000	4.750	3.625	2.208	5.167	3.167	2.083
Frd	.000074						<10 ⁻⁶					
Wp	–	1.000	0.053	1.000	1.000	0.027	–	0.066	0.002	1.000	0.007	0.011
	FDR (Mi)						FNR (Mi)					
Rnk	3.167	2.958	4.333	2.167	3.542	4.833	4.500	3.583	2.292	5.208	3.375	2.042
Frd	.000043						<10 ⁻⁶					
Wp	–	1.000	0.048	1.000	1.000	0.987	–	0.044	0.005	1.000	0.024	0.016
	$T_{\gamma=10,\delta=1}$ (Ma)						$T_{\gamma=10,\delta=1}$ (Mi)					
Rnk	2.750	2.625	4.125	3.708	3.167	4.625	3.250	2.750	4.167	2.417	3.542	4.875
Frd	.001289						.000155					
Wp	–	1.000	0.066	1.000	1.000	0.149	–	1.000	0.139	1.000	1.000	0.884
	Hamming (p value:.0036)											
Rnk	2.750	3.042	4.292	2.458	3.396	5.062						
Frd	.000023											
Wp	–	1.000	0.004	1.000	1.000	0.008						

Rnk stands for average rank over all datasets, 3rd is the p value obtained using the Friedman test and Wp denotes the p value associated with the Wilcoxon test

holds the ordinal numbers of assessed algorithms which are compatible with the numbering introduced in the description of the experimental setup. The second part of the table consists of 11 subsections, and each subsection is related to a quality measure which is presented in the subsection header. Each subsection contains three rows. The first one, denoted by Rnk, shows average ranks achieved by the investigated multi-label classifiers over the benchmark sets. The second row presents p value related to the Friedman test (Frd) applied to criterion-specific results. The last row contains the corrected (Holm's correction) p value (Wp) produced by the Wilcoxon test which was applied to compare the introduced learning procedure against the reference methods.

For easier interpretation, we provided a visualization of the data presented by the above-mentioned tables using radar plots. The plots are shown in Figs. 4, 5, 6 and 7. Each plot presents average ranks achieved by evaluated algorithms under different quality criteria. Additionally, plots also contains a graphical representation of the performed Nemenyi post hoc procedure. Namely, the critical differences are

denoted by black bars parallel to criterion-specific axes in a radar plot.

In order to improve the readability of the paper, tables containing set-specific results for Tversky and zero-one measures are presented in “Appendix”. Results related to the symmetric scenario are shown in Tables 7, 8, 9, 10, 11 and 12, whereas the results for asymmetric scenarios are presented in Tables 15, 16. The detailed description of those tables is provided in the “Appendix”.

5.1 The symmetric loss

Let us begin with the analysis of results related to the first symmetric scenario which is the optimization of the Jacard loss ($\gamma = \delta = 1$). The outcome of the experiments is presented in Tables 3, 7, 8, 9 and Fig. 4.

From the point of view of this paper, we are interested in the lack of significant difference, in terms of Tversky-based loss functions, between the proposed method and the Dembczyński approach. The lack of the statistically

Table 6 Summarized results for the algorithms tailored to optimize the $\mathcal{T}_{1,10}$ loss function

No.	1	2	3	4	5	6	1	2	3	4	5	6
	$T_{\gamma=1,\delta=10}$ (Ex)						Zero-one					
Rnk	3.708	2.792	3.500	4.125	3.083	3.792	2.583	3.812	3.312	5.542	2.771	2.979
Frd	.146046						$<10^{-6}$					
Wp	–	1.000	1.000	1.000	1.000	1.000	–	1.000	0.689	0.003	1.000	1.000
	FDR (Ex)						FNR (Ex)					
Rnk	3.208	3.583	3.583	5.000	2.833	2.792	3.542	2.667	3.542	2.500	4.375	4.375
Frd	.001613						.001613					
Wp	–	1.000	1.000	0.070	1.000	1.000	–	1.000	1.000	1.000	1.000	1.000
	FDR (Ma)						FNR (Ma)					
Rnk	4.125	3.250	3.917	4.542	2.833	2.333	3.708	2.583	2.417	3.000	4.667	4.625
Frd	.001030						.000018					
Wp	–	1.000	1.000	1.000	0.043	1.000	–	1.000	1.000	1.000	1.000	1.000
	FDR (Mi)						FNR (Mi)					
Rnk	4.292	3.625	3.792	4.125	2.792	2.375	3.792	3.083	2.708	1.792	4.750	4.875
Frd	.002742						$<10^{-6}$					
Wp	–	1.000	1.000	1.000	0.110	0.256	–	1.000	1.000	0.000	1.000	1.000
	$T_{\gamma=1,\delta=10}$ (Ma)						$T_{\gamma=1,\delta=10}$ (Mi)					
Rnk	4.417	2.917	3.333	2.625	2.667	5.042	4.083	2.917	3.250	1.792	3.917	5.042
Frd	.000014						$<10^{-6}$					
Wp	–	1.000	1.000	1.000	1.000	1.000	–	1.000	1.000	0.003	1.000	1.000
	Hamming											
Rnk	4.250	3.708	4.000	4.500	2.167	2.375						
Frd	.000018											
Wp	–	1.000	1.000	1.000	0.017	0.256						

Rnk stands for average rank over all datasets, Frd is the p value obtained using the Friedman test and Wp denotes the p value associated with the Wilcoxon test

important difference is confirmed by both statistical tests; however, the average ranks may suggest that the proposed method performs slightly better. This observation leads to a conclusion that the divergence between the F_1 measure and the Jaccard loss is not substantial enough to produce significant difference between the investigated approaches. Additionally, the proposed method introduces an approximation error. The presence of this error has influenced the results.

On the other hand, the proposed method reveals to achieve the best classification quality under the zero-one loss. Only structured SVM approach is statistically comparable to the proposed one; however, the average rank of the proposed method is the lowest. This is an important result because the aforementioned criterion is the most restrictive loss function [33]. What is more, the mentioned measure is also the roughest quality indicator that is unable to distinguish a nearly correct outcome from a totally misclassified one. Thus, significant superiority suggests that our approach achieved the greatest number of 'perfect match' solutions.

Additionally, the experimental evaluation unveils that the proposed approximation procedure tends to be more conservative than the reference methods. In other words, the method reduces the false positive rate at the cost of

increasing the false negative rate, and the imbalance factor is greater in comparison with the other algorithms. This fact is manifested by the low rank earned in terms of FDR and relatively high rank under FNR criteria. This slight shift towards a majority class (the label is irrelevant to given object) must be interpreted as a beneficial property when the label density is low, which is a common situation under the multi-label framework. Moreover, the noticed shift is not a symptom of a harmful bias towards the majority class because the method performs well with respect to the Hamming loss, the Jaccard loss and the zero-one loss.

The results related to the second symmetric scenario ($\gamma = 10, \delta = 10$) are presented in Tables 4, 10, 11, 12 and Fig. 5. The change of the values of the parameters does not induce any major change in the general behaviour of the proposed method. Namely, the algorithm still is the most conservative approach among the investigated procedures.

What is more, it maintains its leading position under the zero-one criterion. An interpretation of this phenomenon may lie in the interpretation of the $T_{\gamma,\delta}$ when the parameters γ and δ are relatively high. Namely, under such conditions, the loss function becomes similar to the zero-one loss. The previously analysed example shows that even the algorithm

Fig. 4 Visualization of average ranks achieved by algorithms and corresponding critical distances for the Nemenyi post hoc test for the algorithms tailored to optimize the $\mathcal{T}_{1,1}$ (Jaccard) loss function. Each axis of the radar plot corresponds to given quality criterion. The closer a point is to the centre of the radar plot, the lower average rank is (lower is better). Algorithms are numbered as in the section that describes experimental setup. Investigated algorithms are also distinguished using different colours and point styles. Lines connecting algorithm-specific points are added only for visualization purpose. Black bars parallel to criterion-specific axes denote critical difference for the Nemenyi tests

Fig. 5 Visualization of average ranks achieved by algorithms and corresponding critical distances for the Nemenyi post hoc test for the algorithms tailored to optimize the $\mathcal{T}_{10,10}$ loss function

Fig. 6 Visualization of average ranks achieved by algorithms and corresponding critical distances for the Nemenyi post hoc test for the algorithms tailored to optimize the $\mathcal{T}_{10,1}$ loss function

Fig. 7 Visualization of average ranks achieved by algorithms and corresponding critical distances for the Nemenyi post hoc test for the algorithms tailored to optimize the $\mathcal{T}_{1,10}$ loss function

focused on optimization of the Jaccard loss outperforms the remaining methods in terms of the zero-one loss.

The quality assessment in terms of the example-based Tversky loss shows that the proposed method significantly outperforms only the BR system and the Jansche classifier. This result is quite obvious if we consider the assumption about the relationship between labels that lie at the root of

Table 7 Results for the algorithms tailored to optimize the $\mathcal{T}_{1,1}$ loss function

No.	$T_{\gamma=1, \delta=1}(\text{Ex})$						Zero-one						$T_{\gamma=1, \delta=1}(\text{Ma})$						$T_{\gamma=1, \delta=1}(\text{Mi})$					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.468	.839	.854	.818	.501	.850	.558	.954	.969	.932	.595	.964	.810	.899	.902	.837	.826	.902	.795	.912	.915	.867	.814	.915
2	.823	.798	.795	.728	.817	.797	1.00	1.00	1.00	1.00	1.00	1.00	.886	.864	.861	.823	.875	.862	.829	.805	.802	.730	.818	.804
3	.822	.829	.839	.836	.909	.862	.983	.995	.996	.989	1.00	.992	.948	.945	.946	.969	.958	.949	.849	.849	.855	.853	.921	.864
4	.823	.831	.840	.843	.911	.865	.985	.996	.995	.991	1.00	.993	.951	.947	.949	.969	.958	.952	.847	.847	.855	.852	.921	.863
5	.819	.828	.837	.840	.908	.863	.980	.994	.994	.991	1.00	.991	.949	.948	.949	.967	.959	.953	.845	.846	.852	.850	.919	.863
6	.436	.439	.819	.546	.884	.455	.468	.480	.917	.602	1.00	.490	.824	.825	.812	.828	.853	.842	.813	.814	.812	.804	.855	.827
7	.465	.467	.469	.479	.472	.471	.732	.788	.796	.771	.831	.794	.529	.511	.511	.542	.504	.510	.519	.511	.512	.517	.506	.511
8	.672	.799	.805	.645	.635	.805	.917	.998	.999	.975	.931	.999	.888	.900	.902	.903	.787	.902	.770	.830	.834	.738	.693	.834
9	.427	.450	.460	.406	.441	.473	.856	.918	.918	.886	.877	.928	.466	.482	.487	.523	.517	.491	.434	.454	.462	.420	.456	.468
10	.259	.255	.903	.239	.206	.256	.285	.282	.941	.258	.217	.282	.836	.833	.877	.807	.793	.831	.856	.859	.879	.862	.931	.855
11	.257	.261	.858	.201	.188	.252	.281	.289	.888	.208	.196	.278	.856	.856	.897	.734	.819	.856	.777	.783	.843	.741	.852	.781
12	.352	.344	.351	.793	.008	.699	.405	.409	.405	1.00	.018	.722	.450	.442	.453	.293	.029	.590	.519	.506	.516	.786	.015	.729
13	.795	.816	.958	.962	.696	.817	.851	.858	1.00	1.00	.998	.943	1.00	.958	.945	.946	.913	.945	.938	.954	.955	.959	.856	.955
14	.911	.916	.917	.781	.569	.917	.996	1.00	1.00	1.00	.998	.943	1.00	.958	.945	.946	.913	.945	.913	.917	.918	.812	.605	.918
15	.511	.527	.526	.912	.263	.629	.620	.648	.642	1.00	.416	.734	.468	.477	.479	.629	.292	.487	.633	.655	.657	.912	.359	.672
16	.530	.560	.563	.562	.635	.567	.711	.822	.827	.715	.919	.828	.630	.630	.631	.635	.694	.633	.640	.639	.639	.646	.687	.640
17	.883	.874	.874	.873	.651	.877	.971	.996	.995	.999	.972	.997	.940	.928	.937	.937	.748	.937	.939	.940	.943	.938	.706	.943
18	.889	.887	.888	.889	.596	.890	.968	.997	.996	.999	.856	.997	.945	.942	.944	.944	.772	.944	.947	.948	.950	.945	.704	.950
19	.892	.888	.889	.907	.595	.891	.971	.998	.997	1.00	.847	.998	.928	.924	.926	.945	.761	.926	.948	.949	.950	.949	.707	.951
20	.484	.487	.876	.787	.933	.485	.498	.505	.957	.842	1.00	.498	.950	.948	.896	.927	.917	.949	.943	.939	.888	.900	.918	.939
21	.449	.527	.527	.527	.419	.548	.538	.813	.812	.746	.614	.831	.572	.585	.586	.565	.509	.591	.582	.607	.609	.581	.526	.614
22	.492	.537	.551	.590	.619	.623	.612	.730	.734	.678	.809	.772	.598	.617	.632	.731	.700	.634	.635	.673	.690	.662	.729	.701
23	.666	.618	.615	.607	.588	.617	.993	.996	.998	.995	.994	.998	.663	.648	.643	.658	.627	.647	.642	.621	.618	.606	.589	.620
24	.580	.574	.576	.484	.670	.580	.892	.902	.907	.850	.991	.905	.692	.685	.684	.717	.702	.688	.625	.621	.620	.524	.686	.624

Zero-one loss and Tversky loss

Table 8 Results for the algorithms tailored to optimize the $\mathcal{T}_{1,1}$ loss function

No.	FDR (Ex)						FNR (Ex)						FDR (Ma)						FNR (Ma)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.428	.807	.821	.785	.476	.818	.394	.641	.653	.686	.389	.649	.724	.884	.888	.802	.766	.888	.678	.539	.527	.610	.666	.529
2	.733	.746	.749	.604	.769	.748	.675	.506	.469	.533	.524	.483	.809	.818	.822	.775	.840	.821	.775	.654	.628	.734	.674	.637
3	.753	.771	.770	.722	.887	.780	.708	.651	.657	.753	.724	.740	.886	.903	.912	.904	.946	.903	.896	.865	.859	.946	.836	.889
4	.753	.771	.770	.724	.889	.784	.710	.656	.667	.769	.730	.750	.887	.906	.913	.912	.947	.907	.900	.868	.865	.946	.837	.892
5	.748	.770	.769	.723	.885	.782	.705	.651	.658	.761	.719	.745	.890	.911	.918	.900	.947	.913	.899	.871	.866	.944	.839	.895
6	.424	.426	.770	.491	.854	.419	.396	.398	.707	.504	.667	.430	.663	.664	.761	.633	.838	.527	.732	.739	.524	.728	.374	.797
7	.381	.415	.421	.377	.426	.423	.301	.226	.223	.320	.184	.227	.404	.429	.433	.388	.436	.431	.311	.230	.226	.351	.199	.229
8	.592	.766	.771	.488	.510	.771	.457	.289	.285	.435	.429	.285	.825	.869	.872	.839	.744	.872	.757	.629	.621	.820	.680	.621
9	.338	.333	.327	.308	.342	.322	.263	.281	.298	.222	.262	.320	.346	.355	.353	.399	.402	.344	.272	.301	.312	.355	.324	.327
10	.256	.253	.890	.237	.202	.253	.231	.228	.876	.224	.200	.231	.783	.782	.864	.676	.689	.791	.684	.684	.491	.751	.767	.689
11	.255	.259	.854	.197	.186	.250	.231	.231	.828	.198	.181	.223	.816	.817	.891	.627	.644	.814	.591	.581	.431	.657	.796	.589
12	.279	.266	.270	.792	.004	.662	.350	.331	.351	.022	.006	.699	.398	.402	.409	.287	.025	.522	.430	.422	.433	.215	.027	.590
13	.791	.815	.958	.961	.664	.816	.400	.160	.272	.446	.633	.130	.705	.764	.771	.818	.559	.771	.518	.423	.399	.593	.558	.399
14	.894	.913	.914	.718	.343	.914	.711	.343	.310	.437	.441	.311	.920	.937	.939	.895	.818	.938	.668	.352	.327	.699	.862	.327
15	.437	.462	.461	.910	.229	.576	.480	.455	.458	.504	.134	.562	.432	.453	.455	.616	.268	.455	.433	.430	.432	.548	.220	.461
16	.476	.514	.516	.506	.608	.520	.395	.325	.324	.461	.302	.330	.500	.540	.542	.441	.615	.542	.416	.348	.346	.482	.322	.352
17	.860	.860	.859	.858	.605	.864	.429	.160	.133	.190	.303	.136	.931	.925	.935	.934	.704	.935	.479	.294	.294	.409	.490	.295
18	.868	.874	.873	.876	.550	.877	.407	.191	.153	.201	.284	.155	.939	.938	.941	.941	.726	.941	.463	.303	.292	.401	.508	.293
19	.873	.875	.875	.899	.552	.878	.390	.177	.146	.202	.289	.148	.922	.921	.924	.942	.713	.924	.433	.287	.277	.411	.499	.277
20	.470	.473	.837	.727	.912	.467	.469	.468	.768	.734	.730	.474	.809	.816	.860	.794	.909	.808	.938	.934	.665	.871	.490	.938
21	.436	.518	.519	.517	.404	.541	.371	.123	.122	.271	.213	.142	.400	.554	.557	.471	.436	.555	.376	.130	.129	.284	.219	.150
22	.454	.512	.522	.559	.594	.598	.410	.314	.314	.524	.372	.445	.508	.573	.593	.571	.663	.559	.458	.365	.362	.653	.449	.430
23	.573	.532	.542	.473	.506	.538	.443	.332	.280	.389	.278	.293	.516	.518	.527	.492	.515	.525	.437	.360	.314	.461	.334	.330
24	.463	.474	.477	.348	.607	.471	.409	.378	.368	.347	.363	.389	.574	.580	.584	.557	.609	.579	.497	.460	.454	.603	.354	.473

FDR and FNR

Table 9 Results for the algorithms tailored to optimize the $\mathcal{T}_{1,1}$ loss function

No.	Hamming						FDR (Mi)						FNR (Mi)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.093	.334	.352	.170	.106	.352	.706	.906	.910	.844	.757	.910	.591	.398	.387	.521	.560	.392
2	.237	.306	.324	.187	.317	.319	.736	.756	.761	.606	.771	.760	.674	.504	.466	.540	.528	.480
3	.030	.036	.037	.026	.059	.030	.754	.785	.797	.720	.899	.772	.717	.660	.665	.763	.729	.747
4	.028	.034	.035	.024	.055	.028	.755	.785	.796	.711	.900	.773	.711	.656	.664	.768	.731	.745
5	.029	.035	.036	.025	.057	.029	.749	.782	.793	.713	.897	.770	.710	.656	.661	.761	.724	.746
6	.146	.144	.258	.145	.447	.125	.639	.632	.767	.627	.841	.534	.723	.728	.505	.709	.373	.785
7	.235	.251	.254	.224	.259	.252	.391	.428	.432	.366	.441	.429	.304	.228	.224	.329	.191	.228
8	.108	.209	.218	.097	.077	.218	.701	.814	.820	.661	.582	.820	.498	.327	.322	.463	.464	.323
9	.283	.297	.300	.273	.305	.298	.319	.331	.329	.306	.336	.320	.233	.260	.276	.221	.253	.295
10	.145	.146	.371	.109	.097	.138	.798	.805	.870	.745	.792	.794	.678	.678	.326	.763	.895	.688
11	.125	.132	.332	.077	.075	.128	.713	.724	.839	.548	.558	.718	.480	.474	.095	.608	.808	.482
12	.030	.029	.029	.172	.001	.034	.282	.274	.270	.784	.005	.000	.410	.397	.414	.026	.011	.729
13	.120	.254	.277	.233	.023	.275	.934	.953	.955	.958	.742	.954	.496	.215	.174	.378	.754	.174
14	.132	.326	.345	.102	.035	.345	.890	.913	.915	.777	.391	.915	.709	.318	.285	.457	.472	.285
15	.025	.028	.029	.155	.013	.025	.438	.511	.513	.903	.273	.441	.487	.459	.461	.486	.156	.554
16	.254	.281	.282	.233	.372	.281	.511	.548	.549	.468	.634	.548	.423	.358	.357	.484	.314	.363
17	.254	.362	.393	.332	.044	.392	.936	.939	.942	.937	.649	.942	.422	.192	.168	.235	.354	.170
18	.273	.371	.399	.337	.040	.398	.945	.947	.949	.944	.646	.949	.415	.221	.191	.254	.355	.193
19	.289	.378	.404	.357	.040	.404	.947	.948	.950	.948	.652	.950	.392	.210	.186	.256	.354	.188
20	.091	.092	.241	.123	.489	.089	.695	.695	.860	.792	.912	.646	.935	.929	.641	.837	.482	.931
21	.156	.241	.243	.178	.154	.242	.444	.582	.585	.498	.450	.585	.373	.131	.130	.285	.226	.150
22	.053	.074	.079	.048	.088	.068	.494	.609	.632	.427	.671	.599	.433	.335	.335	.548	.393	.458
23	.383	.398	.420	.337	.373	.415	.522	.534	.550	.469	.508	.546	.410	.327	.285	.393	.281	.299
24	.294	.303	.308	.215	.415	.303	.487	.500	.506	.354	.614	.500	.418	.388	.378	.356	.371	.399

FDR, FNR and Hamming

the investigated algorithms. Strictly speaking, the outperformed methods, in contrast to the remaining procedures, expect the labels to be conditionally independent, which results in creation of simplified models. As a consequence, the overall, predictive performance of those approaches is generally lower. On the other hand, there are no significant differences between the proposed method and the other methods under label-based Tversky loss functions.

Equally important, the classification quality measured by the example-based Tversky loss has increased. This observation is a positive sign since the proposed method is tailored to optimize the aforesaid quality measure. Furthermore, this fact confirms the previously made conclusion that the predictive ability of the proposed approximation scheme will rise if the disagreement between the Tversky loss and F_1 measure increases. The graphical explanation of this phenomenon is provided by Fig. 1. That is, if the parameters of the Tversky loss become closer to $\{\gamma = 0.5, \delta = 0.5\}$, the curve related to $T_{\gamma,\delta}$ becomes more flat. As a consequence, the linear approximation becomes similar to the curve related to the F_1 loss and the predicted outcome is close to the class assignment provided by the Dembczyński algorithm.

On the other hand, there is still no significant difference between the proposed method and its counterparts in terms of macro-averaged and micro-averaged Tversky measures. Nonetheless, the average ranks show that the classification quality obtained by the Dembczyński algorithm may be a bit better.

5.2 The asymmetric loss

Under the first asymmetric scenario, we set the parameters of the Tversky loss to $\gamma = 10$ and $\delta = 1$, which should result in building classifiers that prefer more conservative solutions in comparison to the outcomes produced by the procedures that employ symmetrical loss measures. Indeed, the obtained results, which are summarized in Tables 5, 13, 14, 15 and Fig. 6, confirmed that assessed procedures exhibit the expected behaviour. To be more precise, the evaluation in terms of the FNR criteria shows a significant decrease in comparison with the binary relevance system which is a reference method that is not affected by the values of parameters γ and δ . The observed cutback concerns all investigated classifiers, even those that were significantly

Table 10 Results for the algorithms tailored to optimize the $\mathcal{T}_{10,10}$ loss function

No.	$T_{\gamma=10,\delta=10}$ (Ex)						Zero-one						$T_{\gamma=10,\delta=10}$ (Ma)						$T_{\gamma=10,\delta=10}$ (Mi)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.511	.935	.950	.912	.559	.946	.527	.954	.969	.932	.574	.964	.863	.987	.986	.941	.952	.987	.971	.990	.991	.985	.976	.991
2	.980	.974	.973	.963	.979	.974	1.00	1.00	1.00	1.00	1.00	1.00	.961	.975	.975	.902	.946	.974	.982	.976	.976	.964	.980	.976
3	.957	.970	.971	.965	.987	.972	.979	.995	.993	.989	.999	.992	.994	.994	.994	.996	.996	.994	.985	.982	.984	.983	.992	.984
4	.957	.971	.971	.968	.988	.973	.980	.996	.993	.991	.999	.993	.994	.994	.994	.996	.995	.994	.984	.982	.984	.983	.992	.984
5	.952	.969	.969	.967	.987	.971	.975	.994	.991	.991	.999	.991	.994	.994	.994	.996	.996	.994	.983	.982	.984	.983	.991	.984
6	.480	.469	.876	.588	.971	.482	.489	.480	.894	.602	.999	.490	.977	.977	.977	.978	.982	.980	.977	.977	.976	.976	.983	.979
7	.666	.722	.727	.714	.731	.728	.712	.788	.794	.771	.794	.794	.916	.905	.905	.909	.908	.904	.921	.912	.912	.914	.914	.912
8	.884	.970	.972	.912	.870	.972	.908	.998	.999	.975	.925	.999	.991	.986	.986	.984	.895	.986	.983	.980	.981	.966	.956	.981
9	.727	.801	.815	.770	.790	.816	.810	.918	.928	.886	.882	.928	.869	.864	.868	.872	.884	.868	.887	.889	.894	.877	.905	.895
10	.276	.277	.934	.254	.209	.277	.279	.282	.941	.258	.211	.282	.953	.947	.984	.914	.826	.947	.987	.982	.986	.983	.994	.982
11	.272	.284	.882	.207	.197	.274	.274	.289	.887	.208	.198	.278	.984	.981	.988	.848	.892	.981	.978	.973	.982	.966	.987	.972
12	.396	.398	.396	.973	.015	.718	.405	.409	.405	1.00	.017	.722	.502	.488	.502	.375	.034	.618	.913	.908	.912	.973	.112	.964
13	.801	.854	.996	.996	.757	.854	.802	.858	1.00	1.00	.766	.858	.626	.785	.793	.847	.612	.793	.996	.995	.995	.996	.985	.995
14	.975	.990	.991	.966	.919	.991	.980	1.00	1.00	.998	.980	1.00	.998	.992	.992	.991	.986	.992	.996	.991	.991	.977	.958	.991
15	.610	.626	.621	.989	.331	.715	.627	.648	.642	1.00	.349	.734	.525	.534	.536	.674	.382	.536	.951	.950	.950	.990	.829	.953
16	.655	.776	.780	.688	.827	.782	.676	.822	.826	.715	.864	.828	.951	.941	.942	.942	.964	.942	.954	.946	.946	.948	.962	.947
17	.943	.979	.978	.982	.878	.980	.953	.996	.995	.999	.928	.997	.995	.985	.992	.992	.907	.992	.992	.994	.994	.993	.962	.994
18	.922	.982	.981	.985	.769	.982	.930	.997	.996	.999	.806	.997	.996	.993	.993	.993	.922	.993	.992	.995	.995	.994	.961	.995
19	.925	.982	.982	.989	.764	.983	.933	.998	.997	1.00	.801	.998	.996	.975	.975	.993	.928	.975	.992	.995	.995	.995	.962	.995
20	.495	.502	.900	.832	.985	.496	.497	.505	.907	.842	1.00	.498	.993	.994	.990	.991	.990	.994	.993	.994	.989	.989	.990	.994
21	.487	.765	.764	.708	.501	.783	.495	.813	.812	.746	.521	.831	.929	.930	.931	.924	.899	.932	.939	.939	.940	.933	.912	.941
22	.565	.697	.671	.663	.710	.746	.579	.730	.698	.678	.734	.772	.807	.815	.816	.897	.834	.820	.949	.954	.955	.951	.961	.959
23	.948	.926	.927	.922	.938	.928	.997	.996	.998	.995	1.00	.998	.956	.944	.943	.943	.946	.944	.956	.942	.942	.939	.947	.942
24	.831	.836	.840	.768	.945	.841	.885	.902	.906	.850	.998	.905	.955	.948	.949	.939	.956	.949	.947	.942	.943	.917	.959	.943

Zero-one loss and Tversky loss

Table 11 Results for the algorithms tailored to optimize the $\mathcal{T}_{10,10}$ loss function

No.	FDR (Ex)						FNR (EX)						FDR (Ma)						FNR (Ma)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.399	.807	.821	.785	.459	.818	.391	.641	.653	.686	.382	.649	.599	.884	.888	.802	.771	.888	.658	.539	.527	.610	.665	.529
2	.736	.746	.749	.604	.777	.748	.720	.506	.469	.533	.559	.483	.798	.818	.822	.775	.843	.821	.785	.654	.628	.734	.668	.637
3	.770	.771	.753	.722	.885	.780	.745	.651	.694	.753	.750	.740	.891	.903	.910	.904	.947	.903	.899	.865	.871	.946	.860	.889
4	.767	.771	.755	.724	.885	.784	.740	.656	.706	.769	.753	.750	.890	.906	.913	.912	.946	.907	.904	.868	.877	.946	.857	.892
5	.759	.770	.755	.723	.882	.782	.733	.651	.700	.761	.745	.745	.898	.911	.918	.900	.947	.913	.907	.871	.879	.944	.859	.895
6	.436	.426	.749	.491	.846	.419	.403	.398	.753	.504	.657	.430	.705	.664	.722	.633	.838	.527	.649	.739	.621	.728	.342	.797
7	.361	.415	.419	.377	.410	.423	.368	.226	.224	.320	.233	.227	.378	.429	.433	.388	.426	.431	.383	.230	.227	.351	.260	.229
8	.656	.766	.771	.488	.487	.771	.653	.289	.285	.435	.455	.285	.874	.869	.872	.839	.731	.872	.851	.629	.621	.820	.699	.621
9	.328	.333	.324	.308	.355	.322	.301	.281	.313	.222	.351	.320	.354	.355	.350	.399	.374	.344	.325	.301	.323	.355	.379	.327
10	.254	.253	.889	.237	.197	.253	.246	.228	.876	.224	.198	.231	.810	.782	.862	.676	.700	.791	.762	.684	.491	.751	.770	.689
11	.258	.259	.854	.197	.190	.250	.253	.231	.829	.198	.186	.223	.823	.817	.892	.627	.655	.814	.645	.581	.436	.657	.821	.589
12	.280	.266	.270	.792	.004	.662	.351	.331	.351	.022	.006	.699	.398	.402	.409	.287	.025	.522	.430	.422	.433	.215	.024	.590
13	.785	.815	.958	.961	.681	.816	.794	.160	.272	.446	.684	.130	.614	.764	.771	.818	.538	.771	.602	.423	.399	.593	.563	.399
14	.918	.913	.914	.718	.557	.914	.899	.343	.310	.437	.382	.311	.930	.937	.939	.895	.872	.938	.912	.352	.327	.699	.746	.327
15	.455	.462	.461	.910	.212	.576	.517	.455	.458	.504	.185	.562	.433	.453	.455	.616	.225	.455	.439	.430	.432	.548	.225	.461
16	.498	.514	.516	.506	.615	.520	.477	.325	.326	.461	.443	.330	.499	.540	.542	.441	.607	.542	.506	.348	.349	.482	.452	.352
17	.850	.860	.859	.858	.580	.864	.855	.160	.133	.190	.453	.136	.894	.925	.935	.934	.665	.935	.914	.294	.294	.409	.600	.295
18	.842	.874	.873	.876	.528	.877	.846	.191	.153	.201	.424	.155	.913	.938	.941	.941	.682	.941	.934	.303	.292	.401	.620	.293
19	.850	.875	.875	.899	.527	.878	.855	.177	.146	.202	.429	.148	.910	.921	.924	.942	.700	.924	.937	.287	.277	.411	.635	.277
20	.472	.473	.807	.727	.899	.467	.467	.468	.843	.734	.707	.474	.807	.816	.831	.794	.899	.808	.922	.934	.840	.871	.407	.938
21	.433	.518	.519	.517	.393	.541	.433	.123	.122	.271	.309	.142	.393	.554	.557	.471	.361	.555	.439	.130	.129	.284	.310	.150
22	.460	.512	.510	.559	.565	.598	.471	.314	.352	.524	.450	.445	.503	.573	.571	.571	.601	.559	.514	.365	.384	.653	.511	.430
23	.573	.532	.540	.473	.582	.538	.553	.332	.284	.389	.266	.293	.518	.518	.526	.492	.574	.525	.542	.360	.319	.461	.293	.330
24	.462	.474	.475	.348	.629	.471	.468	.378	.377	.347	.380	.389	.572	.580	.582	.557	.627	.579	.567	.460	.463	.603	.386	.473

FDR and FNR

Table 12 Results for the algorithms tailored to optimize the $\mathcal{T}_{10,10}$ loss function

No.	Hamming						FDR (Mi)						FNR (Mi)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.062	.334	.352	.170	.100	.352	.597	.906	.910	.844	.739	.910	.667	.398	.387	.521	.558	.392
2	.226	.306	.324	.187	.316	.319	.737	.756	.761	.606	.778	.760	.719	.504	.466	.540	.562	.480
3	.030	.036	.034	.026	.054	.030	.771	.785	.788	.720	.898	.772	.749	.660	.702	.763	.755	.747
4	.028	.034	.032	.024	.051	.028	.768	.785	.788	.711	.897	.773	.739	.656	.702	.768	.754	.745
5	.029	.035	.033	.025	.053	.029	.759	.782	.785	.713	.894	.770	.735	.656	.702	.761	.746	.746
6	.180	.144	.196	.145	.457	.125	.707	.632	.720	.627	.839	.534	.648	.728	.603	.709	.341	.785
7	.229	.251	.253	.224	.251	.252	.366	.428	.431	.366	.425	.429	.374	.228	.225	.329	.244	.228
8	.115	.209	.218	.097	.071	.218	.775	.814	.820	.661	.551	.820	.686	.327	.323	.463	.489	.323
9	.285	.297	.300	.273	.316	.298	.305	.331	.325	.306	.330	.320	.275	.260	.290	.221	.328	.295
10	.144	.146	.362	.109	.097	.138	.821	.805	.868	.745	.807	.794	.792	.678	.326	.763	.919	.688
11	.123	.132	.331	.077	.077	.128	.746	.724	.839	.548	.581	.718	.614	.474	.098	.608	.848	.482
12	.030	.029	.029	.172	.001	.034	.282	.274	.270	.784	.005	.000	.411	.397	.414	.026	.008	.729
13	.025	.254	.277	.233	.019	.275	.915	.953	.955	.958	.683	.954	.938	.215	.174	.378	.810	.174
14	.079	.326	.345	.102	.058	.345	.918	.913	.915	.777	.610	.915	.918	.318	.285	.457	.408	.285
15	.025	.028	.029	.155	.011	.025	.454	.511	.512	.903	.186	.441	.526	.459	.461	.486	.208	.554
16	.256	.281	.281	.233	.351	.281	.517	.548	.548	.468	.634	.548	.503	.358	.359	.484	.447	.363
17	.043	.362	.393	.332	.037	.392	.832	.939	.942	.937	.614	.942	.876	.192	.168	.235	.493	.170
18	.039	.371	.399	.337	.033	.398	.830	.947	.949	.944	.606	.949	.880	.221	.191	.254	.489	.193
19	.038	.378	.404	.357	.033	.404	.825	.948	.950	.948	.609	.950	.884	.210	.186	.256	.491	.188
20	.102	.092	.134	.123	.479	.089	.781	.695	.809	.792	.899	.646	.918	.929	.819	.837	.409	.931
21	.156	.241	.243	.178	.127	.242	.436	.582	.585	.498	.365	.585	.440	.131	.130	.285	.321	.150
22	.051	.074	.072	.048	.069	.068	.474	.609	.606	.427	.606	.599	.496	.335	.371	.548	.473	.458
23	.376	.398	.418	.337	.471	.415	.519	.534	.548	.469	.584	.546	.520	.327	.289	.393	.270	.299
24	.284	.303	.306	.215	.433	.303	.469	.500	.504	.354	.630	.500	.474	.388	.387	.356	.390	.399

FDR, FNR and Hamming

better than BR under the symmetric scenario. Contrary to the results obtained using symmetric quality measures, the significant difference between the proposed method and the Dembczyński approach does not hold although the corresponding p value is close to the assumed significance level. The disappearance of the significant differences can also be observed under FDR criterion. As a consequence, the introduced approach is no longer the most conservative classifier. This phenomenon suggests that the considered asymmetry ratio has pushed algorithms to the point where the precision cannot be further increased without significant deterioration of the recall.

Notwithstanding, our approximation procedure still achieves the lowest average rank in terms of the example-based Tversky loss and zero-one criterion. Generally speaking, the relative differences, which are described by average ranks, fall under the scheme described in the section devoted to the analysis of the first symmetrical scenario.

The analysis of the second asymmetric scenario ($\gamma = 1, \delta = 10$) provides us with interesting information (Tables 6, 16, 17, 18 and Fig. 7). Namely, the attempt to reduce the false negative ratio at the cost of increasing the

number of false positives leads to a significant decrease in the classification quality under each considered loss measure. The decline is confirmed by the Nemenyi post hoc procedure which shows that all classifiers perform in a manner similar to the BR system, although under the symmetric experiment the algorithms were able to outperform the BR system. What is more, the significant differences between the proposed method and reference approaches have not held. The experiment reveals that increasing the δ causes an undesirable rise of the number of labels which are considered relevant. This strategy proves to be inappropriate for the multi-label datasets characterized by the low label density.

6 Conclusions

In this paper, we addressed the issue of building a multi-label classifier that utilizes the Tversky loss. Our approach is built using a discrete linear approximation of the analysed loss function. The optimization procedure was performed using the inner–outer optimization technique proposed by Dembczynski [9] in order to find the optimal solution of the

Table 13 Results for the algorithms tailored to optimize the $\mathcal{T}_{10,1}$ loss function

No.	$T_{\gamma=10,\delta=1}$						Zero-one						$T_{\gamma=1,\delta=10}$ (Ma)						$T_{\gamma=10,\delta=1}$ (Mi)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.473	.918	.935	.882	.545	.931	.516	.952	.969	.919	.574	.964	.805	.982	.983	.926	.909	.983	.936	.990	.990	.982	.967	.990
2	.965	.964	.964	.951	.973	.966	1.00	1.00	1.00	1.00	1.00	1.00	.944	.963	.963	.901	.944	.966	.970	.969	.970	.954	.974	.970
3	.933	.906	.917	.953	.982	.942	.980	.984	.988	.992	.999	.992	.988	.988	.989	.992	.994	.990	.971	.966	.970	.969	.989	.974
4	.935	.908	.919	.956	.982	.943	.981	.986	.989	.994	.999	.993	.989	.989	.989	.992	.994	.990	.971	.966	.970	.968	.989	.974
5	.928	.905	.918	.955	.982	.941	.976	.982	.988	.993	.999	.991	.989	.989	.990	.991	.994	.991	.970	.966	.969	.969	.989	.973
6	.466	.467	.847	.442	.954	.470	.483	.491	.881	.456	.999	.490	.958	.943	.951	.940	.981	.940	.960	.941	.953	.931	.981	.937
7	.582	.643	.679	.830	.707	.703	.710	.744	.771	.907	.816	.794	.859	.866	.872	.879	.871	.874	.864	.873	.882	.886	.886	.886
8	.847	.953	.956	.820	.776	.959	.907	1.00	1.00	.959	.915	.999	.971	.973	.976	.972	.857	.976	.968	.976	.978	.962	.920	.979
9	.658	.657	.697	.685	.675	.718	.810	.918	.928	.964	.892	.928	.779	.740	.765	.819	.805	.773	.819	.808	.823	.794	.827	.833
10	.275	.267	.920	.196	.212	.276	.282	.273	.938	.199	.217	.282	.944	.931	.981	.791	.785	.938	.977	.969	.985	.946	.975	.973
11	.267	.268	.875	.189	.195	.272	.272	.274	.884	.191	.197	.278	.976	.973	.987	.806	.873	.973	.967	.957	.981	.940	.951	.963
12	.355	.354	.351	.973	.013	.699	.405	.408	.405	1.00	.021	.722	.467	.469	.470	.358	.039	.590	.817	.811	.810	.973	.045	.729
13	.793	.853	.995	.996	.740	.854	.801	.858	1.00	1.00	.768	.858	.625	.780	.792	.845	.603	.793	.992	.995	.995	.996	.961	.995
14	.971	.990	.990	.954	.774	.990	.980	1.00	1.00	.996	.958	1.00	.994	.989	.989	.986	.976	.989	.992	.990	.990	.979	.916	.991
15	.537	.560	.565	.993	.298	.673	.621	.639	.638	1.00	.351	.734	.509	.522	.526	.666	.332	.526	.901	.913	.916	.993	.699	.901
16	.592	.683	.730	.806	.802	.754	.654	.759	.804	.840	.863	.828	.905	.907	.918	.880	.940	.920	.918	.919	.925	.877	.945	.927
17	.929	.974	.974	.979	.826	.977	.952	.996	.995	.999	.920	.997	.981	.984	.992	.992	.876	.992	.980	.994	.994	.993	.940	.994
18	.910	.978	.977	.983	.723	.979	.930	.996	.996	1.00	.801	.997	.973	.992	.993	.993	.896	.993	.979	.994	.995	.994	.936	.995
19	.914	.978	.978	.988	.722	.980	.933	.997	.997	1.00	.795	.998	.961	.974	.975	.993	.897	.975	.979	.995	.995	.995	.938	.995
20	.488	.491	.879	.470	.979	.491	.496	.502	.899	.478	1.00	.498	.980	.977	.980	.975	.989	.978	.978	.968	.976	.967	.989	.969
21	.463	.751	.757	.707	.489	.779	.487	.805	.811	.749	.523	.831	.857	.920	.922	.888	.844	.922	.891	.933	.934	.911	.859	.935
22	.514	.552	.613	.752	.674	.729	.563	.605	.663	.771	.720	.772	.766	.773	.788	.836	.805	.799	.903	.914	.929	.858	.939	.940
23	.883	.867	.875	.855	.896	.897	.995	.994	.995	.988	.999	.998	.912	.911	.909	.936	.919	.914	.918	.914	.917	.900	.925	.926
24	.753	.749	.761	.726	.927	.786	.880	.899	.899	.935	.997	.905	.910	.906	.908	.915	.923	.912	.903	.900	.905	.831	.945	.914

Zero-one loss and Tversky loss

Table 14 Results for the algorithms tailored to optimize the $\mathcal{T}_{10,1}$ loss function

No.	FDR (Ex)						FNR (Ex)						FDR (Ma)						FNR (Ma)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.386	.805	.822	.773	.456	.818	.398	.647	.656	.696	.394	.649	.595	.880	.887	.791	.752	.888	.682	.545	.538	.623	.689	.529
2	.744	.736	.739	.605	.780	.748	.750	.632	.591	.832	.562	.483	.793	.810	.814	.812	.845	.821	.795	.738	.710	.855	.670	.637
3	.751	.703	.722	.876	.884	.780	.758	.773	.758	.932	.756	.740	.887	.887	.894	.884	.945	.903	.907	.917	.905	.980	.858	.889
4	.753	.706	.727	.875	.886	.784	.754	.780	.766	.939	.766	.750	.891	.893	.901	.891	.946	.907	.911	.919	.907	.980	.862	.892
5	.745	.705	.727	.875	.881	.782	.747	.775	.763	.939	.755	.745	.895	.890	.903	.878	.948	.913	.913	.920	.909	.981	.864	.895
6	.427	.423	.727	.400	.842	.419	.409	.433	.812	.427	.658	.430	.690	.590	.615	.453	.836	.527	.689	.799	.780	.903	.340	.797
7	.360	.386	.410	.715	.411	.423	.393	.282	.249	.757	.224	.227	.375	.409	.425	.399	.422	.431	.405	.291	.254	.745	.237	.229
8	.650	.747	.760	.484	.456	.771	.665	.307	.294	.602	.504	.285	.838	.862	.870	.850	.710	.872	.861	.651	.631	.860	.718	.621
9	.325	.273	.296	.289	.323	.322	.322	.422	.368	.610	.381	.320	.343	.306	.330	.370	.394	.344	.341	.410	.366	.668	.428	.327
10	.257	.247	.887	.185	.201	.253	.249	.228	.879	.178	.201	.231	.804	.762	.860	.594	.664	.791	.754	.692	.496	.728	.750	.689
11	.257	.251	.852	.184	.189	.250	.254	.234	.831	.185	.188	.223	.820	.805	.889	.604	.696	.814	.650	.609	.452	.748	.825	.589
12	.280	.272	.270	.793	.005	.662	.351	.349	.351	.026	.009	.699	.398	.409	.409	.284	.028	.522	.430	.433	.433	.217	.032	.590
13	.784	.814	.957	.961	.693	.816	.793	.165	.280	.446	.701	.130	.614	.758	.770	.816	.540	.771	.601	.421	.407	.592	.571	.399
14	.919	.910	.911	.749	.420	.914	.899	.371	.328	.617	.417	.311	.931	.936	.937	.887	.848	.938	.909	.379	.349	.717	.815	.327
15	.446	.453	.458	.943	.208	.576	.520	.469	.461	.700	.202	.562	.434	.447	.453	.620	.219	.455	.441	.433	.432	.595	.235	.461
16	.487	.483	.507	.769	.603	.520	.486	.371	.342	.802	.424	.330	.487	.510	.534	.278	.595	.542	.516	.393	.364	.807	.437	.352
17	.848	.858	.859	.859	.555	.864	.863	.163	.133	.198	.485	.136	.878	.924	.935	.934	.646	.935	.914	.296	.294	.410	.635	.295
18	.841	.872	.873	.880	.504	.877	.855	.192	.153	.220	.444	.155	.883	.938	.941	.940	.667	.941	.922	.304	.292	.403	.653	.293
19	.849	.873	.875	.901	.507	.878	.863	.180	.146	.217	.447	.148	.870	.921	.924	.942	.684	.924	.913	.289	.277	.412	.657	.277
20	.468	.466	.797	.453	.902	.467	.472	.479	.863	.462	.705	.474	.827	.788	.801	.703	.900	.808	.939	.940	.896	.954	.419	.938
21	.428	.514	.518	.538	.393	.541	.434	.128	.124	.311	.311	.142	.379	.549	.555	.463	.359	.555	.441	.136	.132	.326	.314	.150
22	.448	.446	.485	.725	.556	.598	.475	.410	.381	.731	.457	.445	.488	.504	.539	.497	.583	.559	.517	.446	.409	.787	.510	.430
23	.555	.494	.504	.441	.537	.538	.601	.516	.415	.804	.326	.293	.500	.489	.495	.596	.536	.525	.596	.556	.459	.853	.373	.330
24	.449	.427	.446	.416	.622	.471	.500	.498	.453	.705	.376	.389	.562	.549	.560	.522	.621	.579	.592	.584	.535	.844	.383	.473

FDR and FNR

Table 15 Results for the algorithms tailored to optimize the $\mathcal{T}_{10,1}$ loss function

No.	Hamming						FDR (Mi)						FNR (Mi)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.059	.323	.349	.161	.096	.352	.573	.904	.910	.844	.739	.910	.680	.408	.395	.557	.596	.392
2	.223	.258	.276	.165	.318	.319	.746	.748	.753	.613	.781	.760	.748	.630	.589	.836	.564	.480
3	.028	.025	.027	.019	.053	.030	.754	.715	.741	.613	.897	.772	.763	.782	.766	.937	.762	.747
4	.026	.023	.025	.018	.050	.028	.753	.716	.743	.596	.898	.773	.754	.779	.764	.939	.767	.745
5	.027	.024	.026	.019	.052	.029	.745	.713	.740	.613	.895	.770	.749	.780	.765	.939	.758	.746
6	.167	.127	.142	.115	.452	.125	.690	.559	.634	.342	.837	.534	.686	.788	.757	.892	.339	.785
7	.232	.235	.247	.271	.252	.252	.364	.396	.419	.321	.428	.429	.399	.285	.250	.755	.230	.228
8	.100	.191	.207	.096	.065	.218	.736	.802	.814	.703	.510	.820	.699	.348	.331	.648	.542	.323
9	.286	.296	.298	.336	.310	.298	.299	.276	.301	.213	.304	.320	.296	.389	.342	.586	.366	.295
10	.142	.125	.355	.085	.098	.138	.817	.769	.867	.578	.776	.794	.779	.683	.336	.817	.904	.688
11	.121	.116	.325	.071	.076	.128	.739	.689	.838	.436	.569	.718	.605	.490	.106	.872	.843	.482
12	.030	.029	.029	.171	.001	.034	.282	.273	.270	.785	.004	.000	.411	.412	.414	.030	.015	.729
13	.025	.245	.270	.233	.018	.275	.914	.952	.954	.958	.667	.954	.937	.224	.185	.379	.832	.174
14	.080	.303	.325	.101	.043	.345	.919	.911	.912	.819	.501	.915	.917	.348	.304	.621	.445	.285
15	.025	.027	.028	.151	.011	.025	.445	.492	.502	.936	.170	.441	.531	.476	.464	.687	.234	.554
16	.250	.259	.275	.214	.338	.281	.506	.519	.541	.221	.621	.548	.514	.404	.375	.811	.431	.363
17	.039	.359	.392	.331	.035	.392	.808	.939	.942	.938	.592	.942	.883	.195	.168	.243	.530	.170
18	.035	.369	.398	.336	.031	.398	.798	.947	.949	.945	.578	.949	.888	.223	.192	.272	.515	.193
19	.035	.375	.404	.357	.031	.404	.798	.948	.950	.948	.583	.950	.891	.213	.187	.268	.517	.188
20	.096	.088	.108	.085	.478	.089	.750	.620	.769	.514	.901	.646	.933	.932	.875	.950	.420	.931
21	.154	.236	.241	.177	.127	.242	.427	.578	.583	.495	.362	.585	.441	.136	.132	.322	.324	.150
22	.049	.053	.062	.044	.067	.068	.455	.496	.555	.233	.593	.599	.500	.434	.401	.753	.481	.458
23	.362	.355	.369	.326	.409	.415	.499	.489	.507	.322	.542	.546	.575	.519	.419	.810	.330	.299
24	.278	.273	.282	.237	.426	.303	.454	.444	.467	.200	.624	.500	.509	.511	.465	.708	.385	.399

FDR, FNR and Hamming

considered classification problem in terms of the F_β loss. Although our algorithm solves the optimization task in a suboptimal manner, it does not put any assumption on the label distribution. Additionally, the Tversky loss allows us to express other loss functions as F_β , Jaccard loss or the zero-one loss. As a consequence, the introduced procedure is more flexible than the other approaches proposed by literature. What is more, the computational cost of employing the modified procedure instead of the original one is relatively low.

During the experimental study, we obtained promising results which clearly show that the posed problem was solved with moderate success. However, there is still room for improvement. Namely, results produced by the proposed approach are comparable (under the Tversky loss) to the outcome of the Dembczynski approach for majority of experimental scenarios. What is more, under the example-based version of the Tversky measure, the average ranks suggest

that the proposed method outperforms the reference method. It also should be noticed that the quality of classification achieved by the proposed procedure was substantially higher when the parameters of the loss function was set in order to emulate the zero-one loss. What is more important, the introduced approximation scheme outperforms the other assessed methods with respect to the zero-one loss.

The presented results lead to cautious optimism, so we are willing to continue the development of the presented approach. Our further studies will be focused on different approximation techniques that are expected to reduce the approximation error. Another branch of our research is aimed at dealing with the imbalanced distribution problem which is a common issue under the multi-label classification framework. This is an important problem because our approach is based on a series of binary-relevance-based estimations of a posterior probability distributions which are strongly affected by the uneven label distribution.

Table 16 Results for the algorithms tailored to optimize the $\mathcal{T}_{1,10}$ loss function

No.	$T_{\gamma=1,\delta=10}(\text{Ex})$						Zero-one						$T_{\gamma=1,\delta=10}(\text{Ma})$						$T_{\gamma=1,\delta=10}(\text{Mi})$					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.702	.877	.891	.910	.532	.886	.758	.955	.969	1.00	.591	.964	.958	.942	.943	.909	.902	.943	.955	.942	.942	.931	.939	.943
2	.919	.892	.885	.811	.942	.915	1.00	1.00	1.00	1.00	1.00	1.00	.943	.926	.921	.802	.905	.937	.926	.904	.898	.816	.944	.925
3	.917	.904	.914	.911	.952	.935	.984	1.00	.998	1.00	1.00	.992	.986	.974	.978	.978	.984	.987	.964	.939	.948	.937	.971	.971
4	.920	.908	.915	.914	.953	.937	.986	1.00	.998	1.00	1.00	.993	.986	.974	.978	.979	.984	.986	.963	.938	.948	.938	.971	.970
5	.916	.905	.912	.912	.952	.935	.983	1.00	.998	1.00	1.00	.991	.986	.974	.978	.978	.984	.987	.963	.938	.947	.937	.970	.970
6	.562	.484	.835	.837	.925	.468	.646	.559	.932	.958	1.00	.490	.895	.946	.892	.874	.907	.973	.896	.949	.891	.873	.911	.974
7	.676	.537	.528	.600	.535	.530	.951	.842	.811	.966	.830	.794	.737	.731	.735	.661	.724	.751	.741	.763	.768	.675	.753	.784
8	.795	.868	.871	.804	.783	.869	.921	.999	.998	1.00	.949	.999	.957	.944	.945	.927	.894	.944	.923	.902	.903	.851	.897	.903
9	.515	.540	.600	.513	.595	.664	.990	.939	.923	.995	.923	.928	.515	.632	.713	.540	.737	.762	.515	.685	.755	.544	.767	.813
10	.290	.261	.916	.571	.207	.259	.332	.292	.947	.615	.214	.282	.919	.904	.916	.867	.798	.904	.922	.940	.916	.924	.982	.954
11	.306	.277	.867	.559	.191	.253	.348	.321	.900	.581	.197	.278	.941	.930	.910	.908	.886	.935	.923	.916	.865	.866	.977	.921
12	.396	.388	.396	.793	.012	.718	.405	.439	.405	1.00	.018	.722	.486	.458	.486	.299	.029	.618	.877	.858	.878	.793	.070	.964
13	.803	.818	.961	.965	.727	.819	.851	.858	1.00	1.00	.790	.858	.715	.771	.773	.839	.624	.773	.960	.958	.959	.966	.968	.958
14	.952	.930	.930	.853	.809	.929	.996	1.00	1.00	1.00	.993	1.00	.985	.967	.966	.953	.956	.967	.970	.937	.937	.869	.879	.936
15	.575	.578	.581	.893	.292	.677	.625	.662	.643	1.00	.432	.734	.497	.501	.506	.673	.357	.518	.908	.896	.904	.921	.664	.929
16	.704	.622	.615	.705	.672	.616	.905	.880	.840	.997	.909	.828	.790	.835	.847	.748	.841	.855	.792	.849	.864	.749	.869	.873
17	.910	.895	.894	.903	.750	.896	.972	.997	.995	1.00	.978	.997	.967	.943	.949	.951	.886	.950	.960	.947	.948	.944	.888	.948
18	.912	.907	.907	.915	.678	.908	.967	.997	.996	1.00	.872	.997	.970	.956	.956	.957	.897	.956	.964	.954	.955	.950	.887	.955
19	.915	.909	.908	.924	.677	.909	.971	.998	.997	1.00	.857	.998	.970	.938	.938	.958	.887	.938	.964	.955	.955	.953	.890	.955
20	.555	.521	.906	.914	.961	.491	.589	.557	.976	.996	1.00	.498	.960	.983	.932	.923	.944	.992	.960	.984	.931	.921	.945	.993
21	.609	.541	.537	.620	.436	.555	.714	.823	.814	.953	.624	.831	.820	.728	.729	.726	.760	.750	.834	.741	.742	.747	.786	.759
22	.537	.681	.605	.693	.654	.647	.642	.895	.770	.902	.819	.772	.756	.738	.747	.847	.807	.762	.892	.845	.861	.854	.895	.908
23	.639	.741	.747	.636	.783	.786	.999	.999	.999	1.00	1.00	.998	.644	.766	.769	.640	.788	.809	.645	.797	.802	.640	.830	.845
24	.696	.684	.693	.634	.815	.744	.993	.929	.923	.981	.998	.905	.712	.850	.861	.703	.892	.901	.705	.824	.837	.675	.888	.884

Zero-one loss and Tversky loss

Table 17 Results for the algorithms tailored to optimize the $\mathcal{T}_{1,10}$ loss function

No.	FDR (Ex)						FNR (EX)						FDR (Ma)						FNR (Ma)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.634	.809	.822	.866	.473	.818	.492	.637	.650	.658	.391	.649	.931	.888	.890	.854	.762	.888	.581	.526	.525	.546	.657	.529
2	.753	.759	.762	.745	.783	.748	.453	.385	.355	.128	.569	.483	.838	.831	.832	.779	.846	.821	.584	.573	.548	.416	.675	.637
3	.763	.846	.814	.855	.886	.780	.691	.479	.553	.460	.710	.740	.888	.928	.930	.944	.947	.903	.890	.776	.804	.818	.837	.889
4	.764	.847	.814	.857	.886	.784	.694	.484	.564	.468	.710	.750	.889	.927	.931	.944	.948	.907	.893	.784	.812	.820	.833	.892
5	.758	.844	.813	.854	.885	.782	.690	.478	.553	.463	.704	.745	.890	.922	.922	.943	.948	.913	.893	.786	.812	.821	.834	.895
6	.558	.469	.793	.808	.847	.419	.349	.375	.625	.564	.650	.430	.826	.736	.801	.819	.832	.527	.280	.624	.299	.202	.346	.797
7	.630	.451	.431	.580	.431	.423	.079	.192	.202	.055	.182	.227	.672	.461	.446	.580	.435	.431	.076	.194	.205	.055	.196	.229
8	.601	.784	.781	.727	.567	.771	.435	.276	.277	.188	.378	.285	.866	.873	.875	.891	.794	.872	.740	.612	.611	.680	.665	.621
9	.515	.393	.359	.485	.381	.322	.000	.159	.242	.030	.247	.320	.515	.412	.374	.510	.419	.344	.014	.188	.245	.090	.275	.327
10	.290	.259	.893	.556	.202	.253	.236	.224	.871	.520	.198	.231	.833	.791	.866	.807	.622	.791	.543	.643	.473	.524	.675	.689
11	.306	.276	.859	.548	.187	.250	.236	.227	.827	.532	.182	.223	.845	.824	.896	.853	.653	.814	.468	.535	.427	.553	.796	.589
12	.279	.269	.270	.791	.005	.662	.350	.300	.351	.013	.005	.699	.398	.402	.409	.291	.025	.522	.430	.406	.433	.204	.021	.590
13	.791	.816	.959	.961	.671	.816	.400	.134	.268	.420	.614	.130	.705	.769	.772	.832	.569	.771	.518	.403	.394	.585	.552	.399
14	.895	.915	.917	.806	.595	.914	.709	.309	.293	.187	.349	.311	.920	.939	.939	.924	.863	.938	.667	.324	.307	.599	.742	.327
15	.447	.476	.465	.888	.235	.576	.475	.430	.454	.234	.118	.562	.432	.458	.457	.659	.292	.455	.431	.423	.432	.459	.229	.461
16	.677	.543	.521	.694	.615	.520	.092	.270	.307	.048	.326	.330	.734	.568	.551	.700	.622	.542	.094	.293	.330	.060	.338	.352
17	.860	.862	.859	.866	.644	.864	.282	.150	.132	.151	.319	.136	.954	.926	.935	.939	.763	.935	.391	.292	.294	.402	.501	.295
18	.867	.876	.873	.883	.583	.877	.276	.177	.153	.159	.298	.155	.958	.939	.941	.946	.772	.941	.383	.299	.292	.394	.510	.293
19	.872	.877	.875	.900	.584	.878	.278	.169	.145	.153	.307	.148	.959	.922	.924	.947	.767	.924	.387	.282	.276	.402	.505	.277
20	.550	.508	.868	.889	.904	.467	.411	.451	.671	.626	.721	.474	.886	.819	.881	.884	.902	.808	.617	.863	.400	.316	.438	.938
21	.600	.524	.520	.611	.410	.541	.166	.119	.120	.100	.212	.142	.752	.560	.559	.609	.437	.555	.168	.126	.127	.106	.219	.150
22	.470	.660	.560	.672	.608	.598	.398	.177	.262	.206	.371	.445	.515	.678	.639	.764	.672	.559	.447	.258	.319	.438	.445	.430
23	.629	.567	.568	.623	.563	.538	.017	.209	.212	.020	.281	.293	.633	.551	.551	.623	.534	.525	.010	.222	.228	.018	.307	.330
24	.681	.516	.503	.603	.626	.471	.016	.251	.279	.039	.380	.389	.694	.607	.601	.678	.622	.579	.014	.324	.353	.160	.387	.473

FNR and FDR

Table 18 Results for the algorithms tailored to optimize the $\mathcal{T}_{1,10}$ loss function

No.	Hamming						FDR (Mi)						FNR (Mi)					
	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
1	.390	.352	.356	.263	.102	.352	.925	.910	.910	.880	.749	.910	.459	.388	.384	.386	.560	.392
2	.415	.363	.380	.403	.318	.319	.808	.768	.771	.746	.784	.760	.454	.382	.352	.130	.572	.480
3	.031	.065	.055	.071	.060	.030	.766	.853	.844	.862	.898	.772	.700	.487	.560	.467	.716	.747
4	.030	.061	.051	.068	.057	.028	.766	.853	.843	.864	.897	.773	.696	.485	.561	.464	.711	.745
5	.031	.063	.053	.070	.059	.029	.760	.850	.840	.861	.896	.770	.695	.483	.555	.463	.708	.746
6	.444	.199	.387	.472	.442	.125	.825	.729	.804	.821	.834	.534	.279	.618	.291	.188	.341	.785
7	.611	.280	.264	.458	.261	.252	.671	.466	.446	.598	.444	.429	.076	.192	.203	.055	.187	.228
8	.135	.224	.228	.172	.093	.218	.753	.823	.826	.758	.640	.820	.470	.312	.314	.205	.409	.323
9	.515	.329	.311	.453	.327	.298	.515	.387	.358	.483	.369	.320	.000	.144	.216	.029	.228	.295
10	.241	.164	.382	.280	.096	.138	.833	.800	.870	.841	.769	.794	.459	.576	.284	.416	.857	.688
11	.257	.164	.341	.216	.076	.128	.845	.772	.843	.781	.570	.718	.401	.441	.092	.208	.809	.482
12	.030	.030	.029	.172	.001	.034	.282	.296	.270	.784	.007	.000	.410	.372	.414	.017	.007	.729
13	.120	.270	.284	.244	.025	.275	.934	.954	.956	.958	.767	.954	.496	.180	.168	.346	.730	.174
14	.133	.349	.366	.164	.059	.345	.890	.916	.918	.814	.612	.915	.707	.284	.267	.181	.361	.285
15	.025	.030	.029	.199	.014	.025	.449	.537	.521	.896	.296	.441	.480	.430	.455	.226	.136	.554
16	.639	.308	.289	.569	.377	.281	.733	.574	.556	.704	.640	.548	.095	.302	.340	.057	.332	.363
17	.416	.370	.393	.343	.050	.392	.952	.940	.942	.936	.687	.942	.286	.182	.168	.188	.364	.170
18	.419	.378	.399	.349	.045	.398	.957	.948	.949	.943	.684	.949	.296	.209	.191	.205	.363	.193
19	.422	.388	.405	.370	.046	.404	.958	.949	.950	.946	.691	.950	.295	.203	.186	.202	.368	.188
20	.301	.123	.425	.509	.479	.089	.885	.803	.883	.888	.903	.646	.619	.856	.383	.274	.438	.931
21	.491	.246	.244	.299	.155	.242	.755	.588	.586	.636	.452	.585	.165	.127	.128	.108	.224	.150
22	.056	.143	.105	.139	.091	.068	.516	.754	.699	.752	.682	.599	.420	.192	.280	.221	.390	.458
23	.624	.457	.462	.589	.432	.415	.634	.571	.574	.620	.557	.546	.009	.207	.212	.014	.273	.299
24	.678	.349	.337	.492	.427	.303	.693	.547	.537	.623	.625	.500	.014	.258	.285	.041	.386	.399

FNR, FDR and Hamming

Acknowledgements The work was supported by the statutory funds of the Department of Systems and Computer Networks, Wrocław University of Science and Technology, under Agreements S50020/K0402. Computational resources provided by PL-Grid Infrastructure.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Appendix 1: Full results

All tables included in this section share common table format. Namely, the header table holds the names of

investigated loss functions and the ordinal numbers of assessed algorithms which are compatible with the numbering introduced in the description of the experimental setup. The second part of the table consists of 24 rows, and each row is related to the set whose number is shown in the first column of the table. Within this part, the set-specific loss, averaged over the CV folds, is shown. Some entries included in this part of the table are set to .000, which means that the average loss related to these entries were less than 10^{-3} . By analogy, entries which are set to 1.000 denotes the values greater than 0.999. The third section contains the average ranks achieved by the investigated multi-label classifiers over the benchmark sets.

Appendix 2: Derivations

Appendix 2.1: Linear approximation

$$\begin{cases} g(K, S, x, b_u(K, S)) = l(K, S, x, b_u(K, S)), \\ g(K, S, x, b_l(K, S)) = l(K, S, x, b_l(K, S)) \end{cases}$$

$$\begin{cases} d(K, S) \sum_{y \in S_S} P(y|x) - c(K, S) \sum_{y \in S_S} P(S|x) b_l(K, S) = \sum_{y \in S_S} P(y|x) - \sum_{y \in S_S} \frac{P(y|x) b_l(K, S)}{\delta S + \gamma K + \eta b_l(K, S)}, \\ d(K, S) \sum_{y \in S_S} P(y|x) - c(K, S) \sum_{y \in S_S} P(S|x) b_u(K, S) = \sum_{y \in S_S} P(y|x) - \sum_{y \in S_S} \frac{P(y|x) b_u(K, S)}{\delta S + \gamma K + \eta b_u(K, S)}, \end{cases}$$

which gives:

$$\begin{cases} c(K, S) = \frac{1}{b_u(K, S) - b_l(K, S)} \left[\sum_{y \in S_S} \frac{P(y|x) b_u(K, S)}{\delta S + \gamma K + \eta b_u(K, S)} - \sum_{y \in S_S} \frac{P(y|x) b_l(K, S)}{\delta S + \gamma K + \eta b_l(K, S)} \right] \\ d(K, S) = 1 - \sum_{y \in S_S} \frac{P(y|x) b_l(K, S)}{\delta S + \gamma K + \eta b_l(K, S)} + c(K, S) b_l(K, S) \end{cases}$$

And consequently:

$$\begin{cases} c(K, S) = \frac{S}{(S + \eta b_u(K, S))(S + \eta b_l(K, S))}, \\ d(K, S) = 1 - \frac{\eta b_u(K, S) b_l(K, S)}{(S + \eta b_u(K, S))(S + \eta b_l(K, S))}. \end{cases}$$

References

- Alvares Cherman E, Metz J, Monard MC (2010) A simple approach to incorporate label dependency in multi-label classification. In: Advances in soft computing. Springer, Berlin, pp 33–43. doi:[10.1007/978-3-642-16773-7_3](https://doi.org/10.1007/978-3-642-16773-7_3)
- Barnard K, Duygulu P, Forsyth D, de Freitas N, Blei DM, Jordan MI (2003) Matching words and pictures. *J Mach Learn Res* 3:1107–1135
- Borg I, Groenen P (1997) Modern multidimensional scaling. Springer, New York. doi:[10.1007/978-1-4757-2711-1](https://doi.org/10.1007/978-1-4757-2711-1)
- Boutell MR, Luo J, Shen X, Brown CM (2004) Learning multi-label scene classification. *Pattern Recognit* 37(9):1757–1771. doi:[10.1016/j.patcog.2004.03.009](https://doi.org/10.1016/j.patcog.2004.03.009)
- Briggs F, Lakshminarayanan B, Neal L, Fern XZ, Raich R, Hadley SJK, Hadley AS, Betts MG (2012) Acoustic classification of multiple simultaneous bird species: a multi-instance multi-label approach. *J Acoust Soc Am* 131(6):4640–4650. doi:[10.1121/1.4707424](https://doi.org/10.1121/1.4707424)
- Chai KMA (2005) Expectation of f-measures. In: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval—SIGIR '05. ACM Press. doi:[10.1145/1076034.1076144](https://doi.org/10.1145/1076034.1076144)
- Cheng W, Hüllermeier E, Dembczynski KJ (2010) Bayes optimal multilabel classification via probabilistic classifier chains. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 279–286
- Dembczynski K, Jachnik A, Kotłowski W, Waegeman W, Hüllermeier E (2013) Optimizing the f-measure in multi-label classification: Plug-in rule approach versus structured loss minimization. In: Dasgupta S, Mcallester D (eds) Proceedings of the 30th international conference on machine learning (ICML-13). JMLR workshop and conference proceedings, vol 28, pp 1130–1138
- Dembczynski KJ, Waegeman W, Cheng W, Hüllermeier E (2011) An exact algorithm for f-measure maximization. In: Shawe-Taylor J, Zemel R, Bartlett P, Pereira F, Weinberger K (eds) Advances in neural information processing systems 24. Curran Associates Inc, New York, pp 1404–1412
- Dembczynski K, Waegeman W, Cheng W, Hüllermeier E (2012) On label dependence and loss minimization in multi-label classification. *Mach Learn* 88(1–2):5–45. doi:[10.1007/s10994-012-5285-8](https://doi.org/10.1007/s10994-012-5285-8)
- Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. *J Mach Learn Res* 7:1–30
- Dimou A, Tsoumakas G, Mezaris V, Kompatsiaris I, Vlahavas I (2009) An empirical study of multi-label learning methods for video annotation. In: 2009 seventh international workshop on content-based multimedia indexing. IEEE. doi:[10.1109/cbmi.2009.37](https://doi.org/10.1109/cbmi.2009.37)
- Diplaris S, Tsoumakas G, Mitkas PA, Vlahavas I (2005) Protein classification with multiple algorithms. In: Bozaniis, Panayiotis, Houstis, Elias N (eds) Advances in informatics. Springer, Berlin, pp 448–456. doi:[10.1007/11573036_42](https://doi.org/10.1007/11573036_42)
- Dez J, Luaces O, del Coz JJ, Bahamonde A (2014) Optimizing different loss functions in multilabel classifications. *Prog Artif Intell* 3(2):107–118. doi:[10.1007/s13748-014-0060-7](https://doi.org/10.1007/s13748-014-0060-7)
- Deroski S, Demar D, Grbovi J (2000) Predicting chemical parameters of river water quality from bioindicator data. *Appl Intell* 13(1):7–17. doi:[10.1023/a:1008323212047](https://doi.org/10.1023/a:1008323212047)
- Elisseff A, Weston J (2001) A kernel method for multi-labelled classification. In: Dietterich TG, Becker S, Ghahramani Z (eds) In advances in neural information processing systems 14. MIT Press, pp 681–687

17. Fiore A, Heer J (2016) http://bailando.sims.berkeley.edu/enron_email.html. Accessed 21 Mar 2016
18. Friedman M (1940) A comparison of alternative tests of significance for the problem of m rankings. *Ann Math Stat* 11(1):86–92. doi:[10.1214/aoms/1177731944](https://doi.org/10.1214/aoms/1177731944)
19. Gantz J, Reinsel D (2012) The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east. Technical report, IDC
20. Gibaja E, Ventura S (2014) Multi-label learning: a review of the state of the art and ongoing research. *Wiley Interdiscip Rev Data Min Knowl Discov* 4(6):411–444. doi:[10.1002/widm.1139](https://doi.org/10.1002/widm.1139)
21. Goncalves EC, Plastino A, Freitas AA (2013) A genetic algorithm for optimizing the label ordering in multi-label classifier chains. In: 2013 IEEE 25th international conference on tools with artificial intelligence. IEEE. doi:[10.1109/ictai.2013.76](https://doi.org/10.1109/ictai.2013.76)
22. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The weka data mining software. *ACM SIGKDD Explor Newsl* 11(1):10. doi:[10.1145/1656274.1656278](https://doi.org/10.1145/1656274.1656278)
23. Hand DJ, Yu K (2001) Idiot's bayes: Not so stupid after all? *Int Stat Rev Rev Int Stat* 69(3):385. doi:[10.2307/1403452](https://doi.org/10.2307/1403452)
24. Heider D, Senge R, Cheng W, Hillermeier E (2013) Multilabel classification for exploiting cross-resistance information in hiv-1 drug resistance prediction. *Bioinformatics* 29(16):1946–1952. doi:[10.1093/bioinformatics/btt331](https://doi.org/10.1093/bioinformatics/btt331)
25. Holm S (1979) A simple sequentially rejective multiple test procedure. *Scand J Stat* 6(2):65–70. doi:[10.2307/4615733](https://doi.org/10.2307/4615733)
26. Jaccard P (1912) The distribution of the flora in the alpine zone. I. *New Phytol* 11(2):37–50. doi:[10.1111/j.1469-8137.1912.tb05611.x](https://doi.org/10.1111/j.1469-8137.1912.tb05611.x)
27. Jansche M (2005) Maximum expected f-measure training of logistic regression models. In: Proceedings of the conference on human language technology and empirical methods in natural language processing—HLT '05. Association for Computational Linguistics. doi:[10.3115/1220575.1220662](https://doi.org/10.3115/1220575.1220662)
28. Jansche M (2007) A Maximum Expected Utility Framework for Binary Sequence Labeling. In: Proceedings of the 45th annual meeting of the association of computational linguistics. Association for Computational Linguistics, Prague, Czech Republic, pp 736–743
29. Jiang JY, Tsai SC, Lee SJ (2012) Fsknn: multi-label text categorization based on fuzzy similarity and k nearest neighbors. *Expert Syst Appl* 39(3):2813–2821. doi:[10.1016/j.eswa.2011.08.141](https://doi.org/10.1016/j.eswa.2011.08.141)
30. Katakis I, Tsoumakas G, Vlahavas I (2008) Multilabel text classification for automated tag suggestion. In: Proceedings of the ECML/PKDD 2008 discovery challenge
31. Kohler M, Krzyzak A (2006) Rate of convergence of local averaging plug-in classification rules under margin condition. In: 2006 IEEE international symposium on information theory. IEEE. doi:[10.1109/isit.2006.261936](https://doi.org/10.1109/isit.2006.261936)
32. Kumar A, Vembu S, Menon AK, Elkan C (2012) Learning and inference in probabilistic classifier chains with beam search. In: Peter A, Flach, De Bie T, Cristianini N (eds) Machine learning and knowledge discovery in databases. Springer, Berlin, pp 665–680. doi:[10.1007/978-3-642-33460-3_48](https://doi.org/10.1007/978-3-642-33460-3_48)
33. Luaces O, Dez J, Barranquero J, del Coz JJ, Bahamonde A (2012) Binary relevance efficacy for multilabel classification. *Prog Artif Intell* 1(4):303–313. doi:[10.1007/s13748-012-0030-x](https://doi.org/10.1007/s13748-012-0030-x)
34. Lichman M (2013) Uci machine learning databases. <http://archive.ics.uci.edu/ml>. Accessed 04 Sept 2017
35. Nan Y, Chai KM, Lee WS, Chieu HL (2012) Optimizing f-measure: a tale of two approaches. In: Langford J, Pineau J (eds) Proceedings of the 29th international conference on machine learning (ICML-12). ACM, New York, NY, USA, pp 289–296
36. Pestian JP, Brew C, Matykiewicz P, Hovermale DJ, Johnson N, Cohen KB, Duch W (2007) 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—BioNLP '07. Association for Computational Linguistics. doi:[10.3115/1572392.1572411](https://doi.org/10.3115/1572392.1572411)
37. Petterson J, Caetano TS (2011) Submodular multi-label learning. In: Shawe-taylor J, Zemel R, Bartlett P, Pereira F, Weinberger K (eds) Advances in neural information processing systems 24, pp 1512–1520
38. Pillai I, Fumera G, Roli F (2013) Multi-label classification with a reject option. *Pattern Recognit* 46(8):2256–2266. doi:[10.1016/j.patcog.2013.01.035](https://doi.org/10.1016/j.patcog.2013.01.035)
39. Pillai I, Fumera G, Roli F (2013) Threshold optimisation for multi-label classifiers. *Pattern Recognit* 46(7):2055–2065. doi:[10.1016/j.patcog.2013.01.012](https://doi.org/10.1016/j.patcog.2013.01.012)
40. Read J, Pfahringer B, Holmes G, Frank E (2011) Classifier chains for multi-label classification. *Mach Learn* 85(3):333–359. doi:[10.1007/s10994-011-5256-5](https://doi.org/10.1007/s10994-011-5256-5)
41. Rijsbergen CJV (1979) Information retrieval, 2nd edn. Butterworth-Heinemann, Newton
42. Rockafellar RT, Wets RJB (1998) Variational analysis. Springer, Berlin. doi:[10.1007/978-3-642-02431-3](https://doi.org/10.1007/978-3-642-02431-3)
43. Sanden C, Zhang JZ (2011) Enhancing multi-label music genre classification through ensemble techniques. In: Proceedings of the 34th international ACM SIGIR conference on research and development in information—SIGIR '11. ACM Press. doi:[10.1145/2009916.2010011](https://doi.org/10.1145/2009916.2010011)
44. Schietgat L, Vens C, Struyf J, Blockeel H, Kocev D, Deroski S (2010) Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinform* 11(1):2. doi:[10.1186/1471-2105-11-2](https://doi.org/10.1186/1471-2105-11-2)
45. Snoek CGM, Worring M, van Gemert JC, Geusebroek JM, Smeulders AWM (2006) The challenge problem for automated detection of 101 semantic concepts in multimedia. In: Proceedings of the 14th annual ACM international conference on multimedia—MULTIMEDIA '06. ACM Press. doi:[10.1145/1180639.1180727](https://doi.org/10.1145/1180639.1180727)
46. Tian Y, Deng N (2005) Support vector classification with nominal attributes. In: Hao Y, Liu J, Wang Y, Cheung YM, Yin H, Jiao L, Ma J, Jiao YC (eds) Computational intelligence and Security. Springer, Berlin, pp 586–591. doi:[10.1007/11596448_86](https://doi.org/10.1007/11596448_86)
47. Trohidis K, Tsoumakas G, Kalliris G, Vlahavas IP (2008) Multi-label classification of music into emotions. In: Bello JP, Chew E, Turnbull D (eds) ISMIR, pp 325–330
48. Tsochantaridis I, Joachims T, Hofmann T, Altun Y (2005) Large margin methods for structured and interdependent output variables. *J Mach Learn Res* 6:1453–1484
49. Tsoumakas G, Katakis I (2007) Multi-label classification. *Int J Data Warehous Min* 3(3):1–13. doi:[10.4018/jdwm.2007070101](https://doi.org/10.4018/jdwm.2007070101)
50. Tsoumakas G, Katakis I, Vlahavas I (2009) Mining multi-label data. In: Data mining and knowledge discovery handbook. Springer, New York, pp 667–685. doi:[10.1007/978-0-387-09823-4_34](https://doi.org/10.1007/978-0-387-09823-4_34)
51. Tsoumakas G, Katakis I, Vlahavas I (2011) Random k -labelsets for multilabel classification. *IEEE Trans Knowl Data Eng* 23(7):1079–1089. doi:[10.1109/tkde.2010.164](https://doi.org/10.1109/tkde.2010.164)
52. Turnbull D, Barrington L, Torres D, Lanckriet G (2008) Semantic annotation and retrieval of music and sound effects. *IEEE Trans Audio Speech Lang Process* 16(2):467–476. doi:[10.1109/tasl.2007.913750](https://doi.org/10.1109/tasl.2007.913750)
53. Tversky A (1977) Features of similarity. *Psychol Rev* 84(4):327–352. doi:[10.1037/0033-295x.84.4.327](https://doi.org/10.1037/0033-295x.84.4.327)
54. Wilcoxon F (1945) Individual comparisons by ranking methods. *Biom Bull* 1(6):80. doi:[10.2307/3001968](https://doi.org/10.2307/3001968)
55. Wu JS, Huang SJ, Zhou ZH (2014) Genome-wide protein function prediction through multi-instance multi-label learning. *IEEE/ACM Trans Comput Biol Bioinform* 11(5):891–902. doi:[10.1109/tcbb.2014.2323058](https://doi.org/10.1109/tcbb.2014.2323058)

56. Xiao X, Wang P, Lin WZ, Jia JH, Chou KC (2013) iamp-2l: a two-level multi-label classifier for identifying antimicrobial peptides and their functional types. *Anal Biochem* 436(2):168–177. doi:[10.1016/j.ab.2013.01.019](https://doi.org/10.1016/j.ab.2013.01.019)
57. Xu XS, Jiang Y, Peng L, Xue X, Zhou ZH (2011) Ensemble approach based on conditional random field for multi-label image and video annotation. In: *Proceedings of the 19th ACM international conference on multimedia—MM '11*. ACM Press. doi:[10.1145/2072298.2072019](https://doi.org/10.1145/2072298.2072019)
58. Yang Y, Joachims T (2008) Text categorization. *Scholarpedia* 3(5):4242. doi:[10.4249/scholarpedia.4242](https://doi.org/10.4249/scholarpedia.4242)
59. Zhang ML, Zhou ZH (2006) Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Trans Knowl Data Eng* 18(10):1338–1351. doi:[10.1109/tkde.2006.162](https://doi.org/10.1109/tkde.2006.162)
60. Zhang X, Song Q (2015) A multi-label learning based kernel automatic recommendation method for support vector machine. *PLoS ONE* 10(4):e0120455. doi:[10.1371/journal.pone.0120455](https://doi.org/10.1371/journal.pone.0120455)
61. Zhou ZH, Zhang ML (2007) Multi-instance multilabel learning with application to scene classification. In: *In advances in neural information processing systems* 19
62. Zhou ZH, Zhang ML, Huang SJ, Li YF (2012) Multi-instance multi-label learning. *Artif Intell* 176(1):2291–2320. doi:[10.1016/j.artint.2011.10.002](https://doi.org/10.1016/j.artint.2011.10.002)