Performance Measures Fusion for Experimental Comparison of Methods for Multi-label Classification

Tome Eftimov
Computer Systems Department
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia

Dragi Kocev
Department of Knowledge Technologies
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia

Abstract

Over the past few years, multi-label classification has been widely explored in the machine learning community. This resulted in a number of multi-label classification methods requiring benchmarking to determine their strengths and weakness. For these reasons, typically, the authors compare the methods using a set of benchmark problems (datasets) with regard to different performance measures. At the end, the results are discussed for each performance measure separately. In order to give a general conclusion in which the contribution of each performance measure will be included, we propose a performance measures fusion approach based on multi criteria decision analysis. The approach provides rankings of the compared methods for each benchmark problem separately. These rankings can then be aggregated to discover sets of correlated measures as well as sets of evaluation measures that are least correlated. The performance and the robustness of the proposed methodology is investigated and illustrated on the results from a comprehensive experimental study including 12 multi-label classification according to 16 performance measures on a set of 11 benchmark problems.

Introduction

Supervised learning is one of the most widely researched and investigated areas of machine learning. The goal in supervised learning is to learn, from a set of examples with known class, a function that outputs a prediction for the class of a previously unseen example. If the examples belong to two classes (e.g., the example has some property or not) the task is called binary classification. The task where the examples can belong to a single class from a given set of m classes ($m \geq 3$) is known as multi-class classification. The case where the output is a real value is called regression.

However, in many real life problems of predictive modelling the output (i.e., the target) can be structured, meaning that there can be more complex output structures such as vectors of variables with some dependencies among them. One type of structured output is vector of binary variables, i.e., the examples can belong to multiple classes simultaneously. This task is known as multi-label classification (MLC). The issue of learning from multi-label data has recently attracted significant attention from many researchers, motivated by an increasing number of new applications. The latter include semantic annotation of images and video (news clips, movies clips), functional genomics (gene and protein function), music categorization into emotions, text classification (news articles, web pages, patents, emails, bookmarks, ...), directed marketing and others. An exhaustive list of multi-label applications is presented in (Gibaja and Ventura 2015).

In recent years, many different approaches have been developed to solving MLC problems. Tsoumakas and Katakis (Tsoumakas and Katakis 2007) summarize them into two main categories: a) algorithm adaptation methods, and b) problem transformation methods. Algorithm adaptation methods extend specific learning algorithms to handle multi-label data directly. Examples include lazy learning (Zhang and Zhou 2007), neural networks (Crammer and Singer 2003), boosting (De Comité, Gilleron, and Tommasi 2003), classification rules (Thabtah, Cowling, and Peng 2004), decision trees (Clare and King 2001) (Blockeel, Raedt, and Ramon 1998) and ensembles thereof (Kocev et al. 2013), ensembles with label subgroups (RAKEL) (Tsoumakas and Vlahavas 2007), ensembles of classifier chains (Read et al. 2011) etc. Problem transformation methods, on the other hand, transform the MLC problem into one or more single-label classification problems. The single-label classification problems are solved with a commonly used single-label classification approach and the output is transformed back into a multi-label representation. The simplest strategies include the one-against-all and one-against-one strategies, also referred to as the binary relevance method (Tsoumakas and Katakis 2007) and pair-wise method (Fürnkranz 2002) respectively.

Performance evaluation for MLC is a more complex task than that of classical single-label classification. Due to the nature of the task: one example can be labelled with multiple labels. Namely, it is difficult to assess which error is worse: two instances with two incorrect labels each or four instances with single incorrect label each. To this end, in any typical multi-label experiment, it is essential to include multiple and contrasting measures because of the additional degrees of freedom that the multi-label setting introduces

Copyright held by the author(s). In A. Martin, K. Hinkelmann, A. Gerber, D. Lenat, F. van Harmelen, P. Clark (Eds.), Proceedings of the AAAI 2019 Spring Symposium on Combining Machine Learning with Knowledge Engineering (AAAI-MAKE 2019). Stanford University, Palo Alto, California, USA, March 25-27, 2019.

CEUR-WS.org/Vol-2350/paper17.pdf

(Madjarov et al. 2012).

The relations among the different evaluation measures in the literature have been theoretically studied and the main findings can be summarized as follows. To begin with, Hamming loss and subset accuracy have a different structure and minimization of one may cause a high regret for the other (Dembczyński et al. 2010). Next, a study on surrogate losses for MLC showed that none of the convex surrogate loss is consistent with ranking loss (Gao and Zhou 2013). Furthermore, the F-measure optimality of the inference algorithm is studied with decision theoretic approaches (Waegeman et al. 2014). Finally, an investigation on the shared properties among different measures yielded a unified understanding for MLC evaluation (Wu and Zhou 2017). All in all, when benchmarking novel MLC methods, it is necessary to compare their performance with existing state-of-the-art methods. However, due to the multitude of evaluation measures, drawing a clear summaries and conclusions is not easy: the methods have different performance compared to the competing methods on the different evaluation measures. This makes proving a summary recommendation a complex task.

Considering this, we propose an approach for experimental comparison of methods for multi-label classification. It is developed for making a general conclusion using a set of user-specified performance measures. For this reason, the approach follows the idea of PROMETHEE methods, which are applicable in different domains such as, business, chemistry, manufacturing, social sciences, agriculture and medicine (Ishizaka and Nemery 2011; Nikouei, Oroujzadeh, and Mehdipour-Ataei 2017). Recently, they were also used in a data-driven approach for evaluating multi-objective optimization algorithms regarding different performance measures (Eftimov, Korošec, and Koroušić Seljak 2018). To the best of our knowledge, they were not used in the domain of MLC. The PROMETHEE methodology works as a ranking scheme for transforming the data for each benchmark dataset instead of using some traditional statistical ranking scheme (e.g., fractional ranking scheme). Further the obtained rankings that are fused from more performance measures are involved in a statistical test to provide a general conclusion from the benchmark experiment.

The main contributions of the paper are:

- A methodology for fusing the various evaluation measures for the task of MLC.
- The proposed methodology is robust considering the inclusion or exclusion of correlated measures.
- We elucidate sets of evaluation measures that should be used together when assessing the predictive performance.
- We identify the correlated measures for each measure separately.

In the reminder of the paper, we first present the proposed method for fusion of the performance measures for MLC. Then, the experimental design is explained followed by the results and discussion. Finally, the conclusions of the paper are presented.

Fusion of performance measures

Let us assume that a comparison needs to be made among m methods (i.e., alternatives) regarding n performance measures (i.e., criteria) on a single multi-label classification problem (i.e., dataset). Let $M = \{M_1, M_2, \ldots, M_m\}$ be the set of methods we want to compare regarding the set of performance measures $Q = \{q_1, q_2, \ldots, q_n\}$. The decision matrix is a $m \times n$ matrix (see Table 1) that contains values of the performance measures obtained for the methods.

Table 1: Decision matrix

	q_1	q_2	$\ldots$	q_n
M_1	$q_1(M_1)$	$q_2(M_1)$	$\ldots$	$q_n(M_1)$
M_2	$q_1(M_2)$	$q_2(M_2)$	$\ldots$	$q_n(M_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
M_m	$q_1(M_m)$	$q_2(M_m)$	$\ldots$	$q_n(M_m)$

For drawing conclusions and making recommendations on methods' usage by considering a set of performance measures, we propose a performance measures fusion approach that follows the idea of PROMETHEE methodology (Brans and Mareschal 2005). More specifically, we exploit the method PROMETHEE II. It is based on making pairwise comparisons within all methods for each performance measure. The differences between the values for each pair of methods according to a specified performance metric are taken into consideration. For larger differences the decision maker might consider larger preferences. The preference function of a performance measure for two methods is defined as the degree of preference of method M_1 over method M_2 as seen in the following equation:

$$P_j(M_1, M_2) = \begin{cases} p_j(d_j(M_1, M_2)), & maximization\ q_j \\ p_j(-d_j(M_1, M_2)), & minimization\ q_j \end{cases}, \quad (1)$$

where $d_j(M_1, M_2) = q_j(M_1) - q_j(M_2)$ is the difference between the values of the methods for the performance measure q_j and $p_j(\cdot)$ is a generalized preference function assigned to that performance measure. There exist six types of generalized preference functions (Brans and Vincke 1985). Some of them require certain preferential parameters to be defined, such as the preference and indifference thresholds. The preference threshold is the smallest amount that is assumed as preference, while the indifference threshold is the greatest amount of difference that is insignificant.

After selecting the preference function for each performance measure, the next step is to define the average preference index and outranking (preference and net) flows. The average preference index for each pair of methods gives information of global comparison between them using all performance measures. The average preference index can be calculated as:

$$\pi(M_1, M_2) = \frac{1}{n} \sum_{j=1}^{n} w_j P_j(M_1, M_2), \quad (2)$$

where w_j represents the relative significance (weight) of the j^{th} performance measure. The higher the weight value of

a given performance measure the higher its relative significance. The selection of the weights is a crucial step in the PROMETHEE II method because it defines the priorities used by the decision-maker. In our case, we used the Shannon entropy weighted method. For the average preference index, we need to point out that it is not a symmetric function, so $\pi(M_1, M_2) \neq \pi(M_2, M_1)$.

To rank the methods, the net flow for each method needs to be calculated. It is the difference between the positive, $\phi(M_i^+)$, and the negative preference flow of the method, $\phi(M_i^-)$. The positive preference flow gives information how a given method is globally better than the other methdos, while the negative preference flow gives the information about how a given method is outranked by all the other methods. The positive preference flow is defined as:

$$\phi(M_i^+) = \frac{1}{(n-1)} \sum_{x \in M} \pi(M_i, x), \tag{3}$$

while the negative preference flow is defined as:

$$\phi(M_i^-) = \frac{1}{(n-1)} \sum_{x \in M} \pi(x, M_i). \tag{4}$$

The net flow of an algorithm is defined as:

$$\phi(M_i) = \phi(M_i^+) - \phi(M_i^-). \tag{5}$$

The PROMETHEE II method ranks the methods by ordering them according to decreasing values of net flows.

Shannon entropy weighted method

To calculate the weights of each performance measure, we use the Shannon entropy weighted method (Boroushaki 2017). For this reason, the decision matrix presented in Table 1 needs to be normalized. Depending of the value that is preferred (smaller or larger), the matrix is normalized using the following equations:

$$q_j(M_i)' = \frac{\max_i(q_j(M_i)) - q_j(M_i)}{\max_i(q_j(M_i)) - \min_i(q_j(M_i))}, \tag{6}$$

or

$$q_j(M_i)' = \frac{q_j(M_i) - \min_i(q_j(M_i))}{\max_i(q_j(M_i)) - \min_i(q_j(M_i))}, \tag{7}$$

where $q_j(M_i)'$ is the normalized value for $q_j(M_i)$. The sums of the performance measures in all methods are defined as

$$D_j = \sum_{i=1}^{m} q_j(M_i)', j = 1, \ldots, n. \tag{8}$$

The entropy for each performance measure is defined as:

$$e_j = K \sum_{i=1}^{m} W\left(\frac{q_j(M_i)'}{D_j}\right), \tag{9}$$

where K is the normalized coefficient defined as:

$$K = \frac{1}{(e^{0.5} - 1)m}, \tag{10}$$

and W is a function defined as:

$$W(x) = xe^{(1-x)} + (1-x)e^x - 1. \tag{11}$$

The weight of each performance measure used in Equation 2 is calculated using the following equation:

$$w_j = \frac{\frac{1}{(n-E)}(1-e_j)}{\sum_{j=1}^{n}\left[\frac{1}{(n-E)}(1-e_j)\right]}, \tag{12}$$

where E is the sum of entropies $E = \sum_{j=1}^{n} e_j$.

Correlation analysis

The existing literature on evaluation methodology for machine learning and especially the ones referring to the task of MLC correctly identify that some of the typically used measures are correlated among themselves. Furthermore, it points out that one needs to consider different uncorrelated measure to get a better insight into the performance of the evaluated methods. To this end, we perform a correlation analysis of the proposed methodology to assess its robustness to correlated measures, and as an additional result we empirically elucidate the correlations among the measures widely used for MLC.

We used a correlation analysis that considers the absolute values of pairwise correlation.Namely, we performed a correlation analysis on each dataset starting by calculating a correlation matrix for each decision matrix presented in Table 1. In our case, the correlation matrix is a $n \times n$ matrix showing Pearson correlation coefficients between the performance measures (Benesty et al. 2009). The Pearson correlation coefficient is a measure of the linear correlation between two performance measures. Its value is between -1 and 1. We then averaged the correlation matrices across datasets. Furthermore, we removed the performance measures that have the average absolute correlation greater than some threshold thus obtaining sets of evaluation measures that are least correlated. Finally, by applying a threshold on the correlation coefficients we obtain the measures that are most correlated among themselves.

Experimental design

The data used to evaluate the performance of the fusion method is taken from (Madjarov et al. 2012). In that study, 12 MLC methods are compared according to a set of 16 performance measures separately. The methods are divided into three groups using the base machine learning algorithm:(1)*SVMs* (BR (Tsoumakas and Katakis 2007), CC (Read et al. 2011), CLR (Park and Fürnkranz 2007), QWML (Mencía, Park, and Fürnkranz 2010), HOMER (Tsoumakas, Katakis, and Vlahavas 2008), RAkEL (Tsoumakas and Vlahavas 2007), ECC (Read et al. 2011), (2) *Decision trees* (ML-C4.5 (Clare and King 2001), PCT (Blockeel, Raedt, and Ramon 1998), RFML-C4.5 (Breiman 2001), RF-PCT (Kocev et al. 2013)), and (3) *Nearest neighbors* (ML-kNN (Zhang and Zhou 2007)).

The evaluation measures of predictive performance are divided into two groups (Madjarov et al. 2012; Tsoumakas and Katakis 2007): *bipartitions-based* and *rankings-based*. The bipartitions-based evaluation measures are calculated based on the comparison of the predicted relevant labels with the ground truth relevant labels. This group of evaluation measures is further divided into *example-based* and *label-based*.

The example-based evaluation measures ((*Hamming loss*, *accuracy*, *precision*, *recall*, F_1 *score* and *subset accuracy*)) are based on the average differences of the actual and the predicted sets of labels over all examples of the evaluation dataset. The label-based evaluation measures (*micro precision*, *micro recall*, *micro* F_1, *macro precision*, *macro recall* and *macro* F_1), on the other hand, assess the predictive performance for each label separately and then average the performance over all labels. The ranking-based evaluation measures (*one-error*, *coverage*, *ranking loss* and *average precision*) compare the predicted ranking of the labels with the ground truth ranking.

Using the set of performance measures, the methods are compared using 11 MLC benchmark datasets: emotions, scene, yeast, medical, enron, corel5k, tmc2007, mediamill, bibtex, delicious, and bookmarks. A detailed explanation of the implementation of the methods, definitions of the performance measures, and the basic statistics of the datasets are given in (Madjarov et al. 2012).

We selected and tested two generalized preference functions defined in Equation 1. First, a usual preference function is used for each performance measure, so we do not need to select the preference and indifference thresholds. The usual preference function is presented in Equation 13. Using this preference function, we can only say if there is a difference or not, but we do not take into account the difference value.

$$p(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases}, \tag{13}$$

Second, a V-shape generalized preference function is used for each performance measure, in which the threshold of strict preference, q, is set to the maximum difference that exists for each preference measure on a given benchmark problem. The V-shape preference function is presented in Equation 14. Using this preference function, all difference values are take into account using a linear function.

$$p(x) = \begin{cases} 0, & x \leq 0 \\ \frac{x}{q}, & 0 \leq x \leq q \\ 1, & x > q \end{cases}, \tag{14}$$

According to the value of each performance measure that is preferable (smaller or larger), the 16 performance measures can be split into two groups: (1)*Minimization* (Hamming loss, One error, Coverage, Ranking loss) and (2) *Maximization* (Precision, Accuracy, Recall, F_1 score, Subset accuracy, Macro precision, Macro recall, Macro F_1, Micro precision, Micro recall, Micro F_1, Average precision).

Results and discussion

We compared the 12 MLC methods using the set of 16 performance measures on each dataset separately by using the performance measures fusion ranking. We performed the analysis for the two preference functions (usual generalized and V-shaped preference generalized function. The latter was used with different threshold of strict preference for each performance measure. The threshold of strict preference for each performance measure was estimated on each dataset separately and it was set as the maximum difference that exists from all pairwise comparisons of the values between the methods regarding the performance measure on that dataset.

The performance measures fusion rankings of the methods obtained using the usual generalized preference function are presented in Table 2, while the rankings obtained using the V-shape generalized preference function are presented in Table 3. Comparing the rankings from the tables, both generalized preference functions yield equal ranking only on the *bookmarks* dataset. The main reason for this is the size of the bookmarks dataset. Namely, most of the methods were not able to return a result given the experimental setting as provided in the study by (Madjarov et al. 2012). This in turn means that the preference functions are calculated on small number of different values for the performance measures (the experiments that did not finish on time were given the equally worst performance as stipulated by (Madjarov et al. 2012)). For all other datasets, the rankings of the methods differ. For example, let us focus on the delicious dataset, for which the rankings only for two methods differ. In the case of usual generalized preference function the RFML-C4.5 is ranked as the second and the RF-PCT is ranked as the first, while in the case of the V-shape generalized preference function they swap their rankings, the RFML-C4.5 is ranked as the first and the RF-PCT as the second. So, to understand why this happens, we will analyze the performance measures fusion approach on the delicious dataset.

When different generalized preference functions are used, it follows that the methods have different net flows. The net flows are dependent from the positive and negative flows, which are related to the average preference index. Furthermore, the average preference index depends from the weights of the performance measures and the selected generalized preference function. In our case, using the Shannon entropy weighted method, the result is that all performance measures are uniformly distributed on each dataset, so they all have the same influence on the end result, $w_j = w, j = 1, \ldots, n$, for both versions of the performance fusion approach. The weight for each performance measure is estimated according to the entropy it conveys. Having this result, it follows that the difference between the rankings in both versions comes from the selection of different generalized preference functions. For this reason, in Figures 1 and 1, the average preference indices, $\pi(RF\text{-}PCT, M_i)$ and $\pi(RFML\text{-}C4.5, M_i)$ used for calculating the positive flows, obtained on the delicious dataset, are presented. Using this figure, we can see that the average preference indices obtained using the usual generalized preference function between the RFML-C4.5 and each of the methods: CLR, QWML, PCT, RAkEL, and ECC, are the same with the average preference indices obtained between the RF-PCT and each of the methods: CLR, QWML, PCT, RAkEL, and ECC. However this is not a case when the V-shape generalized preference function is used. In this case, the same abovementioned average preference indices obtained for RFML-C4.5 are greater than the same average preference indices obtained for RF-PCT.

We inspect the results more closely by inspecting a pair-

Table 2: Performance measures fusion rankings using the usual preference generalized function.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-kNN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	7.00	10.00	8.00	11.00	5.00	3.00	4.00	12.00	6.00	9.00	2.00	1.00
scene	2.00	3.00	5.00	7.00	6.00	11.00	12.00	9.00	1.00	4.00	10.00	8.00
yeast	2.00	4.00	1.00	6.00	3.00	11.00	12.00	8.00	9.00	5.00	10.00	7.00
medical	8.00	5.00	3.00	1.00	2.00	4.00	12.00	10.00	7.00	9.00	11.00	6.00
enron	2.00	7.00	1.00	9.00	4.00	10.00	12.00	11.00	8.00	5.00	6.00	3.00
corel5k	4.00	5.00	1.00	2.00	3.00	9.00	11.00	7.00	12.00	10.00	8.00	6.00
tmc2007	3.00	1.00	4.00	5.00	6.00	11.00	12.00	10.00	7.00	9.00	8.00	2.00
mediamill	4.00	5.00	11.00	10.00	6.00	12.00	7.00	3.00	9.00	8.00	2.00	1.00
bibtex	2.00	1.00	3.00	4.00	5.00	10.00	11.00	8.00	12.00	7.00	9.00	6.00
delicious	4.00	3.00	10.50	10.50	5.00	7.00	8.00	6.00	10.50	10.50	**2.00**	**1.00**
bookmarks	9.00	9.00	9.00	9.00	9.00	2.00	5.00	3.00	9.00	9.00	4.00	1.00

Table 3: Performance measures fusion rankings using the V-shape preference generalized function.

Dataset	BR	CC	CLR	QWML	HOMER	ML-C4.5	PCT	ML-kNN	RAkEL	ECC	RFML-C4.5	RF-PCT
emotions	8.00	10.00	9.00	11.00	5.00	3.00	4.00	12.00	6.00	7.00	2.00	1.00
scene	2.00	1.00	4.00	7.00	6.00	11.00	12.00	8.00	3.00	5.00	10.00	9.00
yeast	2.00	4.00	1.00	9.00	3.00	12.00	11.00	6.00	8.00	5.00	10.00	7.00
medical	10.00	9.00	2.00	1.00	4.00	3.00	12.00	6.00	7.00	8.00	11.00	5.00
enron	1.00	10.00	2.00	7.00	4.00	8.00	12.00	11.00	9.00	6.00	5.00	3.00
corel5k	4.00	5.00	1.00	2.00	3.00	10.00	9.00	7.00	11.00	12.00	6.00	8.00
tmc2007	3.00	2.00	4.00	6.00	5.00	11.00	12.00	10.00	7.00	8.00	9.00	1.00
mediamill	4.00	5.00	12.00	11.00	7.00	10.00	6.00	3.00	8.00	9.00	2.00	1.00
bibtex	2.00	1.00	3.00	4.00	5.00	10.00	11.00	7.00	12.00	8.00	9.00	6.00
delicious	4.00	3.00	10.50	10.50	5.00	7.00	8.00	6.00	10.50	10.50	**1.00**	**2.00**
bookmarks	9.00	9.00	9.00	9.00	9.00	2.00	5.00	3.00	9.00	9.00	4.00	1.00

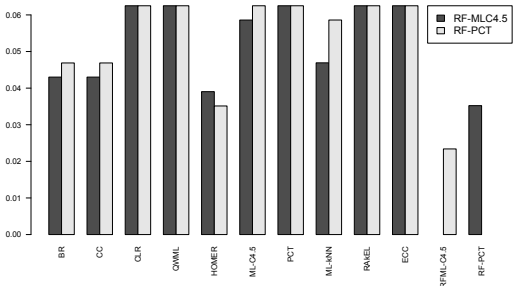

Figure 1: Average preference indices for RFML−C4.5 and RF-PCT obtained on the delicious dataset using the usual preference function ($\pi(RFML\text{-}C4.5, M_i)$ and $\pi(RF\text{-}PCT, M_i)$).

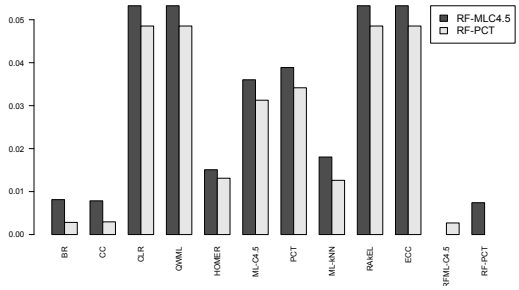

Figure 2: Average preference indices for RFML−C4.5 and RF-PCT obtained on the delicious dataset using the V-shape preference function($\pi(RFML\text{-}C4.5, M_i)$ and $\pi(RF\text{-}PCT, M_i)$).

wise comparison with the ECC method. Using the usual generalized preference function, we can see that $\pi(RFML\text{-}C4.5, ECC) == \pi(RF\text{-}PCT, ECC)$, while if the V-shape generalized preference function is used $\pi(RFML\text{-}C4.5, ECC) > \pi(RF\text{-}PCT, ECC)$. Having the weights uniformly distributed, all of them have the same value, w, the Equation 2 is transformed into:

$$\pi(M_1, M_2) = \frac{1}{n} w \sum_{j=1}^{n} P_j(M_1, M_2). \quad (15)$$

Using the usual generalized preference function, we can see that both methods, RFML−C4.5 and RF−PCT, win against ECC according to all performance measures, but using it we only count wins and losses without taking into account how large are the wins of RFML−C4.5 and RF−PCT against ECC. By using the usual generalized preference function the performance measures fusion approach behaves as majority vote in the case when the influence of each performance measure is uniformly, which happens in our case. However, using the V-shape generalized preference function, the information of how large is the win is also taken into account. Both methods also win against ECC in all performance measures, but here the magnitude of the wins are also considered, which results in different average preference indices. So it follows that RFML-C4.5 ($\sum_{j=1}^{n} P_j(RFML - C4.5, M_2) = 13.63$) has greater average preference index than the average preference index of RF-PCT ($\sum_{j=1}^{n} P_j(RFML - C4.5, M_2) = 12.43$).

After describing the inner working of the proposed method for a single dataset in detail, the obtained rankings for each dataset could be further used with some statistical test to provide a general overall conclusion of the benchmarking of the MLC methods. The Friedman test was selected as an appropriate test for use. The p-value for the rankings obtained with the usual generalized preference function is 0.0005, while the p-value for the rankings obtained using the V-shape generalized preference function is 0.0061. In both cases, the null hypothesis is rejected, so there is a difference between the methods according to the set of 16 performance measures compared on a set of 11 benchmark datasets. To further check where the difference comes from, the Nemenyi post-hoc test (all vs. all) was used with a significance level of 0.05. In the case of usual generalized preference function, the difference come from the pairs of methods (RF-PCT, PCT) and (BR, PCT), while in the case of the V-shape generalized preference function, there is only a difference in the pair (RF-PCT, PCT). This implies that the differences in the rankings of the methods are very small.

We next focus on assessing the robustness of the proposed methodology w.r.t. the presence of correlated measures. Recall that some of the evaluation measures for MLC are correlated among themselves. For this reason, we performed a correlation analysis to investigate whether the method rankings will be disturbed by removing the correlated measures. We performed this analysis using the results from the V-shape generalized preference function. We investigate three predefined correlation thresholds: 0.7, 0.8, and 0.9. The exact values of the thresholds were selected for illustrative purposes. . The performance measures that are not removed for each predefined threshold are (i.e., the least correlated):

- 0.7: coverage, macro precision, micro precision, micro recall, subset accuracy.
- 0.8: hamming loss, macro precision, micro precision, micro recall, precision, ranking loss, subset accuracy.
- 0.9: average precision, hamming loss, macro precision, micro precision, one error, precision, recall, ranking loss, subset accuracy.

The rankings obtained for each predefined threshold are further tested with the Friedman test. In all cases the p-values is smaller than 0.05, so the null hypothesis is rejected and the Nemenyi test was used to get the source of the difference. In all cases there are no big differences in the results from the post-hoc test. When the correlation threshold is set at 0.9, the difference comes from the pairs of methods: (RF-PCT, PCT), (RF-PCT, ECC), and (RF-PCT, RAkEL); in the case of 0.8 from the pairs of methods (RF-PCT, PCT) and (RF-PCT, ECC); and in the case of 0.7 from the pairs of methods (RF-PCT, PCT), (RF-PCT, ECC), and (RF-PCT, RAkEL). If we compare these results with the result obtained when all performance measures are used, there are not big changes, the quesition that arises is only if there is a statistical significance between the pairs (RF-PCT, ECC), and (RF-PCT, RAkEL), which can be further explored within an one vs all analysis.

However, in a lot of papers authors are also interested in the practical significance of the results. The rankings for each method across the datasets for each predefined threshold are thus averaged (Table 4). Next, we check for statistical difference between them using the Friedman test. The p-value is 0.935, so it follows that there is no difference between the average rankings that are obtained for each predefined correlation threshold. Also, for each predefined threshold, we ranked them starting from the best till the worst method according to its average ranking (Table 5). Form here, it follows that there is no big differences regarding the correlation threshold that is used. Notwithstanding, the difference for the HOMER method is noticeable. This is due to the fact that HOMER performs better on the correlated measures (thus its high score). Conversely, CC seems that it performs worse on the correlated measures.

Furthermore, to quantify the robustness, the absolute difference between the rankings obtained on each dataset for each predefined threshold and the rankings obtained using all performance measures are calculated. Next, for each method, the average absolute difference is calculated across datasets to investigate how much the methods change their ranking (Table 6). Using these results, it follows that the rankings are robust to the correlated measures, they can vary, but with a very small differences.

Finally, we use the aggregated correlation matrix across datasets to elucidate the correlated measures. The results are given in Figure 3. The results show a large group of interconnected measures. We can note that *accuracy*, F_1 *score* and *micro* F_1 are connected with most measures (each has 8 connections). The least connected are the ranking based measures.

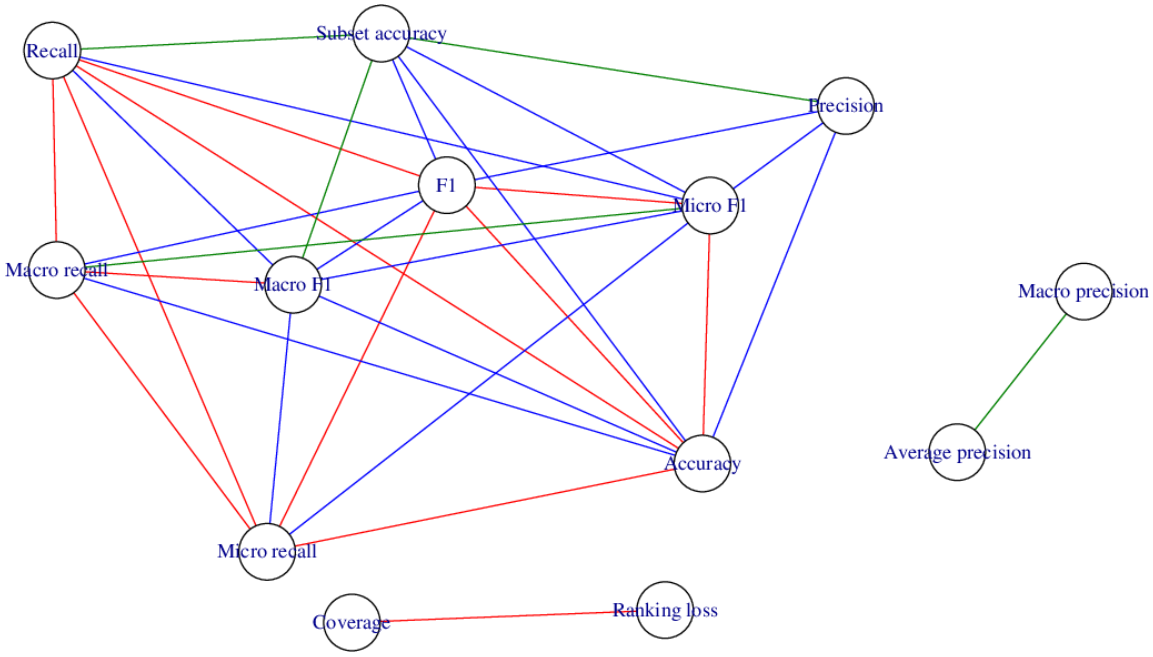

Figure 3: Correlation between performance measures. Red edges are for correlation greater than 0.9, blue and red edges are for correlation greater than 0.8, and green, blue, and red edges correspond to correlation more than 0.7. The evaluation measures *Hamming loss*, *one-error* and *micro precision* are not correlated with the other measures.

Table 4: Average rankings for each method across datasets.

	0.7	0.8	0.9	All
BR	4.82	4.82	4.45	4.45
CC	5.09	5.18	5.45	5.36
CLR	5.14	5.32	5.23	5.23
QWML	6.77	6.77	6.41	7.05
HOMER	6.27	6.55	6.73	5.09
ML-C4.5	8.27	8.09	7.91	7.91
PCT	9.09	9.00	9.27	9.27
ML-kNN	6.91	6.91	6.91	7.18
RAkEL	8.41	7.95	8.50	8.23
ECC	8.59	8.59	8.59	7.95
RFML-C4.5	5.45	5.45	5.36	6.27
RF-PCT	3.18	3.36	3.18	4.00

Table 5: Practical rankings for each method across datasets.

	0.7	0.8	0.9	All
BR	2.00	2.00	2.00	2.00
CC	3.00	3.00	5.00	5.00
CLR	4.00	4.00	3.00	4.00
QWML	7.00	7.00	6.00	7.00
HOMER	6.00	6.00	7.00	3.00
ML-C4.5	9.00	10.00	9.00	9.00
PCT	12.00	12.00	12.00	12.00
ML-kNN	8.00	8.00	8.00	8.00
RAkEL	10.00	9.00	10.00	11.00
ECC	11.00	11.00	11.00	10.00
RFML-C4.5	5.00	5.00	4.00	6.00
RF-PCT	1.00	1.00	1.00	1.00

This is the first attempt at treating the versatile results of MLC experiments in an unified way. More specifically, most of the works in the area report performance along many individual measures and making general conclusions in such a setting is heavily impaired. This is evident also in the extensive experimental comparison performed by (Madjarov et al. 2012), where the results are extensively discussed along multiple evaluation measures. We consider the results from this study to evaluate and illustrate our method because it is the most extensive and most complete study for MLC. We could easily use also other experimental results, but there are not many that follow the same experimental design and have the results readily publicly available.

The potential for practical use of the proposed method is enormous. From a user perspective, the proposed method takes as input the tables with the results does the necessary calculations and outputs the overall rankings of the methods across the different evaluation measures. This is very convenient considering the number of evaluation measures typically used for MLC. This way benchmarking of new methods for MLC can be performed with a great ease. Moreover, it provides the user a nice overview of the methods performance: The proposed methodology shows its robustness on correlated measures and also defines sets of performance measure that are not correlated and can be further included in individual analyses.

We need to mention that the proposed methodology can easily consider also other performance measures such as

Table 6: Average absolute difference between the rankings obtained for each predefined threshold and the rankings obtained using all performance measures across datasets.

	(0.7, All)	(0.8, All)	(0.9, All)
BR	0.91	0.91	0.91
CC	0.64	0.91	1.18
CLR	1.00	1.00	0.55
QWML	0.64	0.64	0.82
HOMER	1.18	1.64	1.64
ML-C4.5	0.73	0.55	0.36
PCT	0.18	0.27	0.00
ML-kNN	0.64	0.64	0.64
RAkEL	0.36	0.82	0.45
ECC	1.00	1.18	1.00
RFML-C4.5	0.82	0.82	0.91
RF-PCT	0.82	0.64	0.82

running times and memory consumption.

Conclusions

In this paper, we propose an approach for fusing multiple evaluation measures for MLC into an overall assessment of performance. The benefit of using this approach is manifold. First, it is designed for making a general conclusion using a set of performance measures. Second, it avoids the comparison according to multiple performance measures separately and then reporting on the results in a biased manner. Third, it is robust to inclusion of correlated evaluation measures. Finally, it gives lists of evaluation measures that are correlated among themselves thus avoiding comparisons only on correlated measures.

For future work, we plan to extend this approach by investigating different preference functions and selecting the best suitable one for each performance measure regarding its properties. Next, we will investigate the building of hybrid methods (mix of more generalized preference functions) that can be used for experimental comparison of methods for MLC. Finally, we will extend the experimental study by including more datasets and methods.

Acknowledgments

This work was supported by the project from the Slovenian Research Agency (research core funding No. P2-0098 and No. P2-0103)

References

Benesty, J.; Chen, J.; Huang, Y.; and Cohen, I. 2009. Pearson correlation coefficient. In *Noise reduction in speech processing*. Springer. 1–4.

Blockeel, H.; Raedt, L. D.; and Ramon, J. 1998. Top-down induction of clustering trees. In *Proceedings of the 15th International Conference on Machine Learning*, 55–63. Morgan Kaufmann.

Boroushaki, S. 2017. Entropy-based weights for multicriteria spatial decision-making. *Yearbook of the Association of Pacific Coast Geographers* 79:168–187.

Brans, J.-P., and Mareschal, B. 2005. Promethee methods. In *Multiple criteria decision analysis: state of the art surveys*. Springer. 163–186.

Brans, J.-P., and Vincke, P. 1985. Note-a preference ranking organisation method: (the promethee method for multiple criteria decision-making). *Management science* 31(6):647–656.

Breiman, L. 2001. Random forests. *Machine learning* 45(1):5–32.

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

Crammer, K., and Singer, Y. 2003. A family of additive online algorithms for category ranking. *Journal of Machine Learning Research* 3:1025–1058.

De Comité, F.; Gilleron, R.; and Tommasi, M. 2003. Learning multi-label alternating decision trees from texts and data. In *Proc. of the 3rd international conference on Machine learning and data mining in pattern recognition*, 35–49.

Dembczyński, K.; Waegeman, W.; Cheng, W.; and Hüllermeier, E. 2010. Regret analysis for performance metrics in multi-label classification: The case of hamming and subset zero-one loss. In *Machine Learning and Knowledge Discovery in Databases*, 280–295. Berlin, Heidelberg: Springer Berlin Heidelberg.

Eftimov, T.; Korošec, P.; and Koroušić Seljak, B. 2018. Data-driven preference-based deep statistical ranking for comparing multi-objective optimization algorithms. In *International Conference on Bioinspired Methods and Their Applications*, 138–150. Springer.

Fürnkranz, J. 2002. Round robin classification. *Journal of Machine Learning Research* 2:721–747.

Gao, W., and Zhou, Z.-H. 2013. On the consistency of multi-label learning. *Artificial intelligence* 199-200:22–44.

Gibaja, E., and Ventura, S. 2015. A Tutorial on Multilabel Learning. *ACM Computing Surveys* 47(3):1–38.

Ishizaka, A., and Nemery, P. 2011. Selecting the best statistical distribution with promethee and gaia. *Computers & Industrial Engineering* 61(4):958–969.

Kocev, D.; Vens, C.; Struyf, J.; and Džeroski, S. 2013. Tree ensembles for predicting structured outputs. *Pattern Recognition* 46(3):817–833.

Madjarov, G.; Kocev, D.; Gjorgjevikj, D.; and Džeroski, S. 2012. An extensive experimental comparison of methods for multi-label learning. *Pattern recognition* 45(9):3084–3104.

Mencía, E. L.; Park, S.-H.; and Fürnkranz, J. 2010. Efficient voting prediction for pairwise multilabel classification. *Neurocomputing* 73(7-9):1164–1176.

Nikouei, M. A.; Oroujzadeh, M.; and Mehdipour-Ataei, S. 2017. The promethee multiple criteria decision making analysis for selecting the best membrane prepared from sulfonated poly (ether ketone) s and poly (ether sulfone) s for proton exchange membrane fuel cell. *Energy* 119:77–85.

Park, S.-H., and Fürnkranz, J. 2007. Efficient pairwise classification. In *European Conference on Machine Learning*, 658–665. Springer.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine learning* 85(3):333.

Thabtah, F. A.; Cowling, P.; and Peng, Y. 2004. MMAC: A New Multi-class, Multi-label Associative Classification Approach. In *Proc. of the 4th IEEE International Conference on Data Mining*, 217–224.

Tsoumakas, G., and Katakis, I. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)* 3(3):1–13.

Tsoumakas, G., and Vlahavas, I. 2007. Random k-labelsets: An ensemble method for multilabel classification. In *European conference on machine learning*, 406–417. Springer.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2008. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD08)*, volume 21, 53–59. sn.

Waegeman, W.; Dembczyński, K.; Jachnik, A.; Cheng, W.; and Hüllermeier, E. 2014. On the bayes-optimality of f-measure maximizers. *Journal of Machine Learning Research* 15:3513–3568.

Wu, K.-Z., and Zhou, Z.-H. 2017. A unified view of multi-label performance measures. In *Proceedings of the 28th International Conference on Machine Learning*, ICML'17.

Zhang, M.-L., and Zhou, Z.-H. 2007. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition* 40(7):2038–2048.