Jordanian Journal of Computers and Information Technology (JJCIT), Vol. 06, No. 04, December 2020.

MULTI-LABEL RANKING METHOD BASED ON POSITIVE CLASS CORRELATIONS

Raed Alazaidah[1], Farzana Kabir Ahmad[2], Mohamad Farhan Mohamad Mohsin[2] and Wael Ahmad AlZoubi[3]

(Received: 19-Jun.-2020, Revised: 8-Aug.-2020, Accepted: 30-Aug.-2020)

ABSTRACT

Multi-label classification is a general type of classification that has attracted many researchers in the last two decades due to its applicability to many modern domains, such as scene classification, bioinformatics and text classification, among others. This type of classification allows instances to be associated with more than one class label at the same time. Class label ranking is a crucial problem in multi-label classification research, because it directly impacts the performance of the final classifiers, as labels with high ranks get a higher chance of being applied. This paper presents a new multi-label ranking algorithm called Multi-label Ranking based on Positive Correlations among labels (MLR-PC). MLR-PC captures positive correlations among labels to reduce the large search space and assigns the true rank per class label for multi-label classification problems. More importantly, MLR-PC utilizes novel problem transformation methods that facilitate exploiting accurate positive correlations among labels. This improves the predictive performance of the classification models derived. Empirical results using different multi-label datasets and five evaluation metrics reveal that the MLR-PC is superior to other commonly existing classification algorithms.

KEYWORDS

Prediction, Machine learning, Multi-label ranking, Multi-label classification, Problem transformation methods, Class ranking methods.

1. INTRODUCTION

Classification is a vital task in supervised learning that has attracted many researchers in the last few decades [1]. Classification learns rules for allocating instances to a class from a "training set" that has explicit classes. It then classifies new instances according to those rules. The accuracy of classification can be assessed by doing this to a "test set" for which the classes are known, but are not used in the classification [2].

In general, according to [3], classification problems are divided into two main categories: single-label classification (SLC) and multi-label classification (MLC). The former necessitates one class label per training instance, while the latter allows multiple class labels per instance. Thus, class labels in the SLC problems are always mutually exclusive [4], whereas class labels in MLC are not. Labels in MLC possibly have some kind of correlation [5].

In MLC problems, the task of Label Ranking (LR) is essential. It reveals the significance and the worthiness of each class label in the prediction phase. Hence, allocating each class label to its true rank is crucial. A common way to accomplish LR is to rank the available class labels according to their frequencies or probabilities [6].

One main challenge of MLC problems is the large problem search space, especially when there are large numbers of class labels and high-dimensional datasets [3]. For example, when the MLC problem contains 20 class labels, then the problem search space consists of 220 possibilities, which is computationally not cost-effective. Hence, cutting down the search space becomes a requirement.

A number of promising research attempts have been conducted in the last few years to reduce the large search space of MLC problems (i.e., [7]-[9]). These approaches dealt with MLC problems through capturing and exploiting the correlations among labels. However, many of these research studies suffer from drawbacks, mainly the limited extent of the type of correlations among class labels being captured

1. R. Alazaidah, is with Computer Science Department, Faculty of Science and IT, Jadara University. Email: raeddiab@yahoo.com
2. F. Ahmad, and M. Mohsin are with School of Computing, Universiti Utara Malaysia, Malaysia. Emails: farzana58@uum.edu.my, Farhan@uum.edu.my
3. W. Alzoubi is with Applied Science Department, Ajloun Uni. College, Al-Balqa Applied Uni., Jordan. Email: wa2010@bau.edu.jo

and adopting inefficient search techniques when capturing the correlations [10].

In this paper, a new MLR algorithm is proposed that utilizes novel problem transformation methods and reveals the positive pairwise correlations among existing labels. Considering the positive pairwise correlations among labels as a transformation criterion will facilitate the capturing and exploiting of the most accurate high-order correlations among labels. In addition, to build classification models, the proposed algorithm integrates class association rules derived by the predictive association rule mining algorithm to determine significant positive correlations among class labels. This process ensures that any negative correlations among classes are discarded (more details are given in sub-section 3.1).

The rest of the paper is organized as follows: Section 2 reviews the literature relevant to MLC, while Section 3 presents the proposed algorithm. Section 4 discusses the evaluation of the proposed algorithm and finally, Section 5 concludes and recommends future work.

2. LITERATURE REVIEW

MLC is a challenging problem that has attracted several scholars in the last two decades. At first, it was motivated by two domains: text classification [11] and medical diagnosis [12]-[13]. After that, MLC has been applied in several other domains, such as: automatic image and video annotation [14]-[16], classification of songs according to the emotions they invoke [17], gene functionality detection [18]-[20], protein functionality detection [21]-[22], drug discovery [23], social network mining [24]-[25], direct marketing [26] and Web mining [27].

Two main approaches have been utilized in dealing with the problem of MLC. The first approach attempts to fit a multi-label dataset into a single label classifier by transforming the multi-label dataset into one or more single-label datasets [28]. This approach has been known as PTM. The second approach adapts a single-label classifier to handle a multi-label dataset and is called the Algorithm Adaptation Method (AAM). According to [29], PTMs are preferable over AAMs, because they are easier to understand and are not domain-specific. This paper deals with the MLC problem from a PTM perspective.

Several PTMs can be found in the literature such as simple selection transformation methods. Simple selection transformation methods transform a multi-label dataset into a single-label dataset based on using the frequency of labels as a transformation criterion [30]. Hence, a multi-label instance can be transformed to be linked with the Most Frequent Label (MFL) or the Least Frequent Label (LFL). Other simple selection transformation methods ignore any multi-label instance or simply choose one of the labels that are associated with an instance randomly. This has been stated by several researchers as one of the best ways to reduce the large problem search space of the MLC problem [7], [9], [31], [52]. Therefore, the proposed algorithm in this research adopts simple PTMs that are based on positive pairwise correlations among labels, which is expected to maximize the utilization of the most significant positive correlations among labels (See Section3 for further details).

In [51], the authors questioned the usefulness of using simple transformation methods that are based on label frequency. Therefore and in order to maximize the exploitation of the most accurate positive dependencies among labels, they proposed three novel simple problem transformation methods based on the positive dependencies among labels and not based on the frequency of labels as in the traditional transformation methods. The first transformation method has been dubbed HAPCF, short for High Accurate Positive Correlation First, where the pairwise positive correlations for the labels are captured and then, the labels are ordered in a descendent way according to the accuracy of the high accurate positive correlation for each label. The second transformation method is called High Standard Deviation First (HSDF), where the Standard Deviation for the accuracy of the pairwise positive correlations for each label is calculated and then, the labels are ordered in a descendent way. The third transformation method is a hybrid method of the first and the second method and has been called High Accurate Positive Correlation and Standard Deviation First (HAPCSDF).

The proposed transformation methods have been extensively evaluated using seven different multi-label datasets and five evaluation metrics, where they showed a superior performance compared with the existing transformation methods. The authors concluded that utilizing the correlations among labels as a transformation criterion is better than using the frequency of labels as a transformation criterion. Furthermore, according to the degree of the captured correlations among labels, MLC algorithms could

be categorized into three types. The first type is known as the first-order approach, which ignores any correlations among labels. Hence, labels in the first-order approach are considered as mutually exclusive. This approach has the advantage of being simple, but suffers from low predictive performance, especially with large datasets that have a high number of labels [32]. Examples of the first-order MLC algorithms are: the Binary Relevance and the Multi-label K Nearest Neighbor (ML-KNN) algorithm [33].

The second type is called the second-order approach and depends on extensive pairwise comparisons among labels while considering features values [3]. This approach suffers from a limited ability to capture correlations among labels as well as the substantial number of the pairwise comparisons that are needed. Thus, the second approach is unsuitable for datasets with a high number of labels [3]. Examples of the second-order approach algorithms are: the Ranking by Pairwise Comparisons (RPC) algorithm [34] and the Calibrated Label Ranking (CLR) algorithm [35].

The third type of MLC algorithm according to the degree of correlations being captured is the high-order approach. This approach captures high-order correlations among labels in the whole dataset or among a subset of the dataset [32]. Usually, this approach suffers from a high-complexity issue due to utilizing complex techniques to capture the correlations among labels. Nevertheless, the high-order approach tends to be better than the previously discussed two approaches, especially for datasets with high cardinality [3]. Examples of high-order approach algorithms are: Label Powerset (LP) [36], Pruned Set (PS) and Ensemble of Pruned Set (EPS) [37], RAKEL [38], Classifier Chains (CC) and Ensemble of Classifier Chains (ECC) [29] and MLC-ACL [30].

Several algorithms that belong to different learning strategies have been proposed to solve the problem of MLC. In [36], an algorithm called HOMER was presented. HOMER is short for Hierarchy of Multi-label classifiERs. HOMER aims to handle large datasets by using a tree structure. It is a divide-and-conquer-based algorithm that constructs a tree recursively in a top-down, depth-first fashion, starting from the root. The HOMER algorithm has been evaluated using two large datasets (delicious and mediamill) and compared with the BR method. HOMER outperforms BR in prediction accuracy, running time and scalability to large datasets. However, HOMER needs to be evaluated against different algorithms and methods and not only the BR method. The complexity of HOMER could be reasonable when applied to large datasets, but HOMER will be inefficient when applied to small or moderate datasets. HOMER is more suitable to large datasets with a large number of labels.

Zhang and Wu (2015) [39] questioned the usefulness of using the same traditional feature selection methods with MLC and, based on their reflections, they proposed an algorithm that focused on extracting label specific features. The algorithm has been named as Multi-label Learning with Label-specific Features (LIFT). LIFT starts by applying clustering techniques on each label to determine its positive and negative instances. Then, features that are specific to each label in the label set are constructed by using the positive instances that were found in the first step. Finally, (k) classifiers are used in the training step. Each classifier is trained using the specific features that were generated previously and for every label in the label set.

LIFT was evaluated using eight datasets from different domains and compared to several state-of-the-art algorithms. The evaluation process concluded that the effectiveness of using new feature-selection techniques was more suitable to the nature of the MLC problem. LIFT has the advantage of being a general approach that could be used with any multi-label algorithm as a preprocessing step that may enhance the effectiveness and the efficiency of the algorithm. On the other hand, LIFT ignores any correlations among labels in the process of selecting the features.

Back Propagation for Multilabel Learning (BP-MLL) algorithm [26] is an adaptation of the traditional multi-layer, feed-forward neural network to multi-label data. The net was trained with a gradient descendent and error back propagation with an error function closely related to the ranking loss that took into account the multi-label data. Experimental results showed a competitive performance in genomics and text categorization domains, with a computational cost derived according to neural network methods. Rokach, Schclar and Itach (2014) [40] questioned the usefulness of selecting the subsets randomly in RAKEL. Their view was based on the idea that dividing the original label sets into smaller subsets should be considered wisely and not randomly. These subsets should reserve the inter-label correlations and other constraints. The chosen (k) subsets should cover all labels and be the minimum possible. The authors proposed using approximation algorithms to determine the size and contents of

the subsets. They proved the efficiency of their algorithms by using different evaluation metrics and different datasets. The only limitations of their work were high complexity and running time [41].

LR methods handle the problem of Multi-label Learning (MLL) by transforming it into a problem of ranking, where pairwise comparisons are performed among all labels and based on these comparisons, a final ranking is obtained. Two main popular methods that could be found in the literature of MLL that are based on pairwise comparisons. The first method is called Ranking by Pairwise Comparisons (RPC) [34]. RPC is similar to BR in dividing a dataset with (k) labels into (k (k-1)/2) binary datasets; a binary dataset for each pair of labels (L1, L2), where the instances of the dataset are those instances that are associated with L1 or L2, but not both labels. To classify a new instance, all the binary models are invoked and a ranking is obtained by counting the votes for each label. RPC suffers from several limitations, such as high quadratic complexity that makes it a very bad choice when dealing with a large number of labels. The last limitation of RPC is that it does not have a split point between relevant and irrelevant labels [3].

CLR method is another pairwise method that enhanced RPC by introducing a calibration label. This virtual label (L0) works as a split point between relevant labels and irrelevant labels [35]. As in RPC, the CLR method suffers from space complexity and computational complexity as well [3].

3. THE PROPOSED ALGORITHM: MULTI-LABEL RANKING ALGORITHM BASED ON POSITIVE HIGH-ORDER CORRELATIONS AMONG LABELS (MLR-PC)

The MLR-PC algorithm comprises three main phases: the transformation phase, the multi-label classifier construction phase and the prediction phase. In the first phase, MLR-PC transforms a multi-label dataset into a single-label dataset using a transformation method based on the positive pairwise correlations among labels and applies a rule-based classifier on the transformed dataset to construct a single-label classifier. The PART algorithm [42] has been chosen in this paper as a base classifier. In the second phase, a multi-label classifier is constructed based on the discovered high-order positive correlations among labels that respect the transformation order of the labels. The last phase involves assigning the predicted ranked labels to a test instance. The main steps of the MLR-PC algorithm are shown in Algorithm 1 and more details are given in the following subsections.

Input: D - Multi-label dataset, *minacc* – minimum accuracy threshold
Output: mlC - multi-label Classifier
Building Model (D, *minacc*, TD)
{
1 Algorithm2 (D)
2 For each x ∈X in TD // X = The set of all labels
3 {
4 Generate all Positive Association Rules (PARs) in a form <<x ⟶ y>>, where y has a lower transformation order than x, using Predictive Apriori algorithm. / /with respect to *minacc*
5 Repeat the previous step having (x and y) in the Antecedent and z in the Consequent, where z ∈λ-{x,y}
6 PARs ⟵ Algorithm3 (PAR)
7 PARs ⟵ Merge (PAR), where the Antecedent is x and the consequent belongs to (λ-x)
8 For each rule in S and have x in its consequent, Replace (x, PARs)
9 }
10 Return (mlC)

Algorithm 1. MLR-PC algorithm.

3.1 Transformation Phase

The transformation phase fits the multi-label dataset into the single-label classifier and often this step depends on the frequency of labels as a transformation criterion. For the proposed algorithm, the transformation phase relies on a new criterion that is based on the positive pairwise correlations among labels. Hence, the label space of the input multi-label dataset is first extracted and then class association rules are derived using the Predictive Apriori algorithm [43]. These rules are utilized to capture the positive pairwise correlations among labels by keeping only positive rules in the form of "IF C1=1 THEN C2=1"for further analysis regardless of the accuracy measure of the rules. In the proposed algorithm, the multi-label dataset is transformed into a single-label dataset based on one of the following transformation methods: (HAPCF, HSDF, HAPCSDF, MFL and LFL). For more clarification and information regarding the previous three PTMs, the reader is advised to read reference [51]. The

following step (Step 9) aims to formulate the transformed Single-label Dataset (SLD) using one of the previous PTMs. After that, PART algorithm is trained on the formulated SLD to construct the single-label classifier (S).

The transformation phase has been performed based on the positive pairwise correlations among labels and not based on the frequency of labels. Algorithm 2 shows the transformation algorithm that has been adopted in the MLR-PC algorithm.

Input: D - Multi-label dataset, minacc – minimum accuracy threshold
Output: S - Single label classifier
Transformation Phase (D)
{
1 TD ⟵ Extract Label Space (D) // TD refers to the label space, where it is represented as a transactional dataset
2 For each item (x) in TD
3 {
4 PARs set ⟵ Predictive Apriori (TD) // regardless the predictive accuracy of PARs
5 HAPCF ⟵ Predictive Accuracy of the highest PAR
6 HSDF ⟵ Compute the Standard Deviation among the Predictive Accuracy of the captured PARs
7 HAPCSDF ⟵ HAPCF + HSDF
8 }
9 SLD ⟵ Transform (D, {HAPCF, HSDF, HAPCSDF}) //SLD: the transformed Single Label Dataset
10 S ⟵ PART (SLD) / with respect to *minacc* // S= single label classifier
11 Return (S, TD)

Algorithm 2. Transformation algorithm.

After transforming the input multi-label dataset into a single-label dataset, classifiers, such as PART, JRip, BayesNet and RIPPER, among others, could be used as a base classifier for the transformed dataset (SLD). In this paper, the PART algorithm has been selected as a base classifier after a thorough evaluation of different algorithms. Table 1 shows the main characteristics of the multi-label datasets considered in this paper.

Table 1. Dataset characteristics.

Dataset	Instances	Attributes	Labels	LCard	Domain
Yeast	2417	103	14	4.327	Biology
Scene	2712	294	6	1.074	Image
Emotions	593	72	6	1.868	Media
Flags	194	19	7	3.392	Image
Genbase	662	1186	27	1.252	Biology
TMC2007	28596	500	22	2.16	Text
Ohsumed	13929	1002	23	1.66	Text

To determine the best base classifier to use in MLR-PC algorithm, five different classifiers have been evaluated on the datasets shown in Table 1. These classifiers are: BayesNet [44], JRip [45], Simple Logistic [46], Decision Table [47] and PART [42]. Table 2 shows the evaluation of several rule-based single-label classifiers, on four different multi-label datasets, based on the Accuracy metric.

Two main points could be inferred from Table 2. The first point is that the transformation method being used has a significant influence on the accuracy rates of the base classifier. In general, the MFL method shows a better accuracy on three datasets (Emotions, Flags and Yeast), while LFL shows a better accuracy on the Scene dataset, which has a low cardinality that is nearly 1. The second point is that the PART algorithm achieved the best performance in terms of accuracy on all datasets regardless of the transformation method being used. Therefore, the PART algorithm has been used as a base classifier for the proposed MLR-PC algorithm.

3.2 Multi-label Classifier Construction Phase

Step 2 to Step 8 of the MLR-PC algorithm (Algorithm 1) produce multi-label rules by integrating the single-label rules (s) discovered by the PART algorithm into multi-label rules. This goal is achieved by capturing the high-order positive correlations among labels based on the positive pairwise correlations discovered earlier by Algorithm 2. For example, a positive pairwise correlation exists between labels

C1, C2 and C3, then MLR-PC attempts to discover a rule like "IF C1 and C2 THEN C3" or a rule like "IF C1 THEN C2 and C3", where C2 and C3 must have a transformation order less than the transformation order of C1. To make the building multi-label rules phase clear, suppose that the following rules have been discovered with label X in the antecedent:

Table 2. The predictive accuracy of several classifiers on the transformed versions of some multi-label datasets.

Base Classifier	Emotions		Flags		Yeast		Scene	
	MFL	LFL	MFL	LFL	MFL	LFL	MFL	LFL
BayesNet	72.770	63.860	75.380	41.530	71.900	30.100	77.170	82.600
JRip	82.670	72.280	81.540	46.150	80.940	45.480	78.850	78.930
Simple Logistic	73.760	73.760	81.530	52.300	71.900	73.240	80.850	86.780
Decision Table	69.800	57.430	75.380	41.540	71.910	30.770	58.860	63.380
PART	**96.530**	**96.040**	**89.230**	**84.620**	**95.650**	**92.310**	**98.410**	**98.830**

IF X THEN Y Accuracy (0.887)
IF X THEN Y and Z Accuracy (0.659)
IF X THEN Y and W Accuracy (0.742)

The above three rules are ordered and merged into a single rule using Algorithm 3 as follows: IF X THEN Y, W, Z. Algorithm 3 shows the sorting procedure for the captured positive association rules. After that and for all single-label rules learned by PART algorithm, every rule that has label X as a consequent will be modified by the MLR-PC algorithm. Thus, the new consequent is Y, W, Z. The process of converting all single-label rules learned by PART continues with all other rules in the same way.

Input: Set of positive association rules

Output: Sorted positive association rules

For any two given rules, r_1 and r_2, r_1 precedes r_2 if:

1. The Predictive Accuracy of r_1 is higher than that of r_2.
2. Both rules have the same Accuracy value, but the cardinality of r_1 is higher than that of r_2.
3. Chose randomly, when the two previous conditions are the same for r_1 and r_2.

Algorithm 3. Ordering the positive association rules algorithm.

3.3 Prediction Phase

As a test instance is about to be classified, the prediction procedure of MLR-PC works as follows. The procedure starts with searching all over the final multi-label rules in the rule set (mlC), to find the best rule that matches the instance test (the rule's body matches some attribute values of the test instance). As the best rule that matches the instance is determined, the set of the labels of that rule is associated with the test instance in the same order as they appear in the consequent of the fired rule. This method utilizes only one rule to associate the predicted class label to a test instance.

4. EMPIRICAL ANALYSIS OF MLR-PC ALGORITHM

The MLR-PC algorithm has been implemented using Java and integrated into the Waikato Environment for Knowledge Analysis (WEKA) software system [48], which is an open source Java software. All experiments were carried out on a Pentium IV, Core i3, 2.10 GHz computer. The training datasets and the testing datasets were chosen according to dataset author recommendation, where nearly two thirds of the datasets have been used as training sets and one third of them been used as testing sets. All datasets are available in Mulan, a multi-label dataset repository [49].The following two sub-sections show the evaluation of the proposed algorithm. Sub-sections 4.1 and 4.2 discuss the evaluation of the MLR-PC algorithm on regular-sized and large-sized datasets, respectively.

4.1 Evaluation of the Proposed MLR-PC Algorithm on Regular-Sized Datasets

Table 3 to Table 6 show a comparison between the proposed MLR-PC algorithm and other MLL algorithms. The compared algorithms have been chosen to represent the three main MLL approaches. The first-order approach was represented by two algorithms: BR and ML-KNN. The second-order approach was represented by two algorithms: BP-MLL and CLR. Finally, the high-order approach was represented by seven algorithms: LP, RAKEL, CC, PS, ECC, EPS and ML-LOC.

Also, the chosen algorithms belong to both PTMs (BR, CLR, LP, RAKEL, CC, PS, ECC and EPS) and AAMs (ML-KNN and BP-MLL). Five multi-label evaluation metrics have been used to evaluate the proposed MLR-PC algorithm: Accuracy, Hamming Loss, Exact Match, One-error and Coverage [50], [53].

Accuracy measures the percentage of those labels that were correctly predicted, with respect to the total number of labels and averaged over all instances. Accuracy is computed using the following equation:

$$Accuracy \quad = \frac{1}{t}\sum_{i=1}^{t}\frac{|Z_i \cap Y_i|}{|Z_i \cup Y_i|} \tag{1}$$

Hamming Loss is a multi-label classification metric that measures how many times on average an instance-label is misclassified. This metric considers both error predictions (when the wrong label is predicted) and omission errors (when the correct label is not predicted). For this metric, the lower the value, the better the accuracy and the performance of the classifier [51]. Hamming Loss is computed using the following equation, where (Δ) denotes the symmetric difference between the grounded truth label set and the predicted set.

$$Hamming\ loss = \frac{1}{t}\sum_{i=1}^{t}\frac{1}{q}|Z_i \triangle Y_i| \tag{2}$$

Exact Match is a restrict metric that does not distinguish between partially correct and completely incorrect prediction. This metric calculates the average of instances whose predicted labels are exactly the same as their grounded truth labels. Exact Match is computed using the following equation and must be maximized:

$$Exact\ Match = \frac{1}{t}\sum_{i=1}^{t}[\,Z_i = Y_i] \tag{3}$$

One-error metric calculates how many times the top–ranked label was not in the set of predicted labels. For this metric, it is clear that it is not suitable for MLL problem; since it considers only the top–ranked label and neglects all other labels. One-error metric must be minimized and is calculated using the following equation:

$$One - error = \frac{1}{t}\sum_{i=1}^{t}[\,arg\,min\,\tau_i(\lambda) \notin Y_i, \lambda \in \mathcal{L}] \tag{4}$$

Coverage measures the average depth in the ranking, in order to cover all the labels associated with an instance. The lower the value of the Coverage metric, the better the accuracy and the performance. This metric is more suitable than the One-error metric for MLL problem; since it considers all labels associated with the instance and not only the top–ranked label. The following equation is used to calculate the Coverage metric:

$$Coverage = \frac{1}{t}\sum_{i=1}^{t} max\,\tau_i(\lambda) - 1\,, \lambda \in Y_i \tag{5}$$

It is worth mentioning that (Y_i) represents the grounded truth label set, while (Z_i) represents the predicted label set. Also, (t) and (q) represent the total number of instances and the total number of labels in the dataset, respectively.

Four regular-size multi-label datasets have been considered in the evaluation of the proposed MLR-PC algorithm. Table 3 shows the evaluation of the proposed MLR-PC algorithm, using five PTMs (HAPCF, HSDF, HAPCSDF, MFL and LFL), based on the Accuracy metric. Table 3 shows clearly that the MLR-PC algorithm has the best accuracy among all other considered algorithms.

Table 3. Accuracy rates of the different MLL algorithms on regular-sized datasets.

	Algorithm	Yeast	Scene	Emotions	Flags
MLR-PC + PTM	MLR-PC-HAPCF	0.532	**0.908**	**0.738**	**0.671**
	MLR-PC-HSDF	**0.583**	**0.908**	**0.738**	0.617
	MLR-PC-	0.538	**0.908**	**0.738**	0.620
	MLR-PC-LFL	0.514	0.881	0.718	0.562
	MLR-PC-MFL	0.280	0.885	0.559	0.483
1st order	BR	0.520	0.643	0.551	0.576
	ML-KNN	0.520	0.691	0.366	0.555
2nd Order	BP-MLL	0.185	0.212	0.276	NG
	CLR	0.514	0.695	0.557	NG
High Order	LP	0.530	0.735	0.584	NG
	RAKEL	0.493	0.694	0.592	NG
	CC	0.521	0.736	0.584	NG
	PS	0.533	0.751	0.599	NG
	ECC	0.299	0.270	0.282	NG
	EPS	0.537	0.751	0.599	NG
	ML-LOC	0.510	NG	0.497	0.568

Table 4. Hamming Loss rates of the different considered MLL algorithms on regular-sized datasets.

	Algorithm	Yeast	Scene	Emotions	Flags
MLR-PC + PTM	MLR-PC-HAPCF	0.144	**0.001**	**0.116**	**0.174**
	MLR-PC-HSDF	**0.127**	**0.001**	**0.116**	0.187
	MLR-PC-HAPCSDF	0.143	**0.001**	**0.116**	0.187
	MLR-PC-LFL	0.158	**0.001**	0.119	0.213
	MLR-PC-MFL	0.219	**0.001**	0.149	0.266
1st order	BR	0.193	0.009	0.188	0.274
	ML-KNN	0.193	0.008	0.262	0.284
2nd order	BP-MLL	0.322	0.057	0.433	NG
	CLR	0.226	0.101	0.214	NG
High Order	LP	0.206	0.090	0.198	NG
	RAKEL	0.207	0.095	0.186	NG
	CC	0.211	0.100	0.197	NG
	PS	0.205	0.084	0.192	NG
	ECC	0.619	0.470	0.630	NG
	EPS	0.207	0.085	0.193	NG
	ML-LOC	0.193	NG	0.210	0.262

For the Scene dataset, no positive correlations among labels were discovered. Nevertheless, the accuracy of the MLR-PC algorithm on this dataset is still the highest. The reason for that is the high accuracy of the base classifier being used (PART). Also, for the Emotions dataset, same transformation orders have been discovered, when using any of the correlation-based PTMs. Hence, MLR-PC has the same accuracy using any of the correlation-based PTMs. Finally, Table 3 demonstrates that the correlation-based PTMs have greatly affected the accuracy of the proposed MLR-PC algorithm.It is worth mentioning that "NG" refers to a "Not Given" value, either because the metric is not applicable to the algorithm or it has not been provided in the original article. Also, all algorithms and methods have been considered with their default settings as stated in their original articles.

Table 4 shows the evaluation results of the proposed MLR-PC algorithm, with respect to the PTMs being used, using the Hamming Loss metric. The table shows clearly that the MLR-PC algorithm has the lowest Hamming Loss of all the algorithms. Also, the performance of the proposed MLR-PC algorithm when utilizing the correlation-based PTMs (HAPCF, HSDF and HAPCSDF) is much better than when utilizing the conventional frequency-based PTMs (MFL and LFL) on most multi-label datasets.

Figure 1 shows the Exact Match results of the proposed MLR-PC algorithm, with respect to the PTMs being used, compared with other algorithms. Figure1 shows that the proposed MLR-PC algorithm has the best Exact Match on the Emotions, Scene and Flags datasets, while PS has the best Exact Match on the Yeast dataset. In general, the proposed MLR-PC algorithm shows an excellent predictive performance, with respect to the Exact Match metric on most regular-sized datasets considered in this paper.

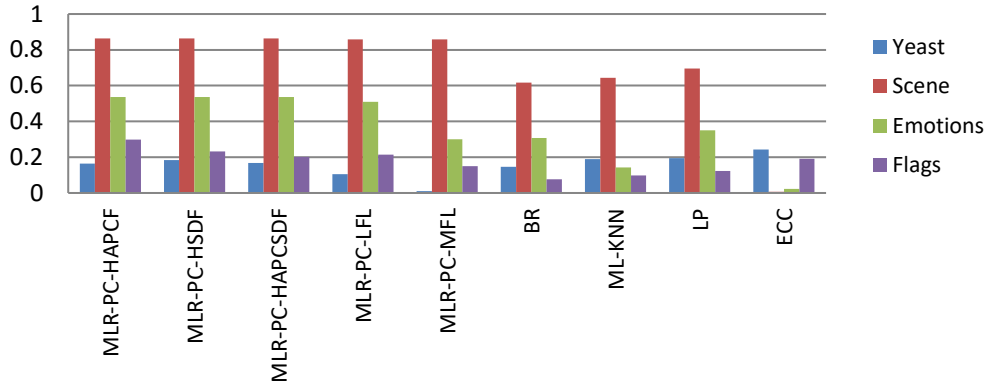

Figure 1. Evaluation of the proposed MLR-PC algorithm using the Exact Match metric on regular-sized datasets.

Table 5. One-error rates of the different considered MLL algorithms on regular-sized datasets.

	Algorithm	Yeast	Scene	Emotions	Flags
MLR-PC + PTM	MLR-PC-HAPCF	0.053	**0.055**	0.039	**0.076**
	MLR-PC-HSDF	0.076	**0.055**	0.039	0.107
	MLR-PC-HAPCSDF	0.063	**0.055**	0.039	0.107
	MLR-PC-LFL	0.076	0.061	0.039	0.153
	MLR-PC-MFL	**0.043**	0.059	**0.034**	0.107
1st order	BR	0.227	0.262	0.256	NG
	ML-KNN	0.228	0.219	0.263	NG
2nd order	BP-MLL	0.235	0.821	0.318	NG
	CLR	0.241	0.323	0.291	NG
High Order	LP	0.267	0.246	0.310	NG
	RAKEL	0.255	0.237	0.260	NG
	CC	0.256	0.268	0.283	NG
	PS	0.321	0.287	0.427	NG
	ECC	0.685	0.775	0.802	NG
	EPS	0.265	0.225	0.300	NG
	ML-LOC	NG	NG	NG	NG
	BR+	NG	NG	NG	NG

Table 5 shows the evaluation results of the proposed MLR-PC algorithm, with respect to the PTMs being used, using the One-error metric. Table 5 shows clearly that MLR-PC algorithm has the best results among all other MLL algorithms, considering the One-error metric. The main reason for

that is using PART algorithm as a base classifier. PART algorithm shows a superior performance on multi-label datasets that usually have special characteristics, such as high dimensionality, high number of instances and most attributes being continuous.

Table 6 shows the evaluation of the proposed MLR-PC algorithm considering the Coverage metric with respect to the PTMs being used. It is obvious from Table 6 that MLR-PC algorithm has the best Coverage among all MLL algorithms considered in this research on the four regular-sized datasets.

Table 6. Coverage rates of the different considered MLL algorithms on regular-sized datasets.

	Algorithm	Yeast	Scene	Emotions	Flags
MLR-PC + PTM	MLR-PC-HAPCF	5.172	**0.133**	**1.193**	2.892
	MLR-PC-HSDF	4.597	**0.133**	**1.193**	2.938
	MLR-PC-HAPCSDF	5.107	**0.133**	**1.193**	**2.830**
	MLR-PC-LFL	**4.298**	0.158	1.352	3.000
	MLR-PC-MFL	5.650	0.162	1.524	3.738
1st order	BR	6.350	1.232	2.400	NG
	ML-KNN	6.300	0.456	2.320	NG
2nd order	BP-MLL	8.005	0.744	3.150	NG
	CLR	NG	NG	NG	NG
High Order	LP	8.065	0.733	2.235	NG
	RAKEL	9.155	0.593	1.986	NG
	CC	7.249	0.619	1.756	NG
	PS	8.313	0.845	2.331	NG
	ECC	10.731	2.662	3.817	NG
	EPS	8.303	0.689	2.138	NG

4.2 Evaluation of the Proposed MLR-PC Algorithm on Large-Sized Datasets

Table 7 shows the evaluation results of the proposed MLR-PC algorithm, using five PTMs (HAPCF, HSDF, HAPCSDF, MFL and LFL), based on the Accuracy metric, with respect to several other algorithms. Table 7 clearly shows that the proposed MLR-PC algorithm has a superior performance on the three large-sized datasets. Also, the proposed MLR-PC algorithm has the best accuracy values when considering the correlation-based PTMs, especially on the Genbase dataset.

Table 8 shows the evaluation results of the proposed MLR-PC algorithm on large-sized datasets, considering the Hamming Loss metric, with respect to several MLL algorithms. From Table 8, the conclusion can be made that the proposed MLR-PC algorithm has the best Hamming Loss metric on the three large-sized datasets, especially when using the correlations-based PTMs.

Table 7. Evaluation of the proposed MLR-PC algorithm using Accuracy metric on large-sized datasets.

	Algorithm	Genbase	TMC2007	Ohsumed
MLR-PC + PTM	MLR-PC-HAPCF	0.985	**0.654**	**0.741**
	MLR-PC-HSDF	**0.988**	**0.654**	**0.741**
	MLR-PC-HAPCSDF	0.987	**0.654**	**0.741**
	MLR-PC-LFL	0.981	**0.654**	**0.741**
	MLR-PC-MFL	0.929	0.635	**0.741**

1st order	BR	0.962	0.541	0.361
	ML-KNN	0.948	0.531	0.355
2nd order	BP-MLL	0.632	0.652	0.403
	CLR	0.561	0.506	0.374
High Order	RAKEL	0.982	0.549	0.383
	LIFT	NG	NG	NG
	ECC	0.978	0.517	0.426
	EPS	0.945	0.549	0.424

Table 8. Evaluation of the proposed MLR-PC algorithm using the Hamming Loss metric on large-sized datasets.

	Algorithm	**Genbase**	**TMC2007**	**Ohsumed**
MLR-PC + PTM	MLR-PC-HAPCF	**0.001**	**0.039**	**0.002**
	MLR-PC-HSDF	**0.001**	**0.039**	**0.002**
	MLR-PC-HAPCSDF	**0.001**	**0.039**	**0.002**
	MLR-PC-LFL	0.002	**0.039**	**0.002**
	MLR-PC-MFL	0.008	0.043	**0.002**
1st order	BR	**0.001**	0.071	0.007
	ML-KNN	0.005	0.073	0.007
2nd order	BP-MLL	0.004	0.098	0.008
	CLR	0.004	0.068	0.008
High Order	RAKEL	0.003	0.068	0.043
	LIFT	0.003	NG	0.056
	ECC	0.002	0.068	0.067
	EPS	0.007	0.069	0.074

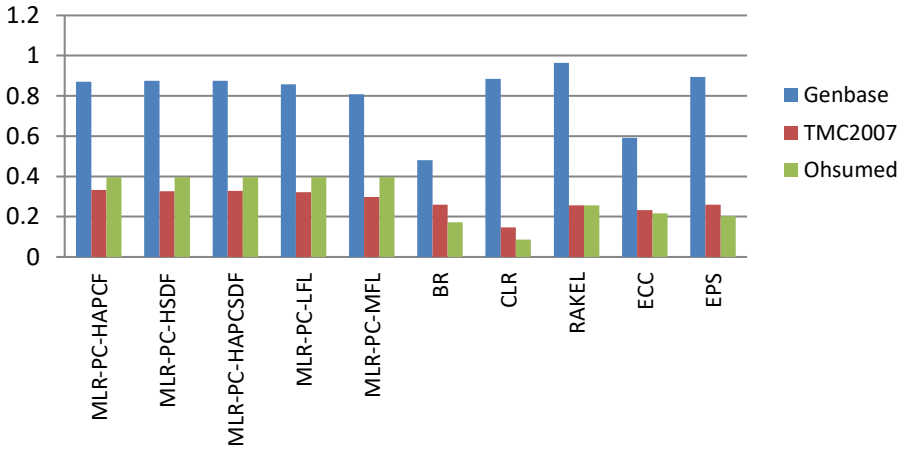

Figure 2.Evaluation of the proposed MLR-PC algorithm using the Exact Match metric on large-sized datasets.

Figure 2 shows the evaluation results of the proposed MLR-PC algorithm using the Exact Match metric, with respect to other MLL algorithms. Figure 2 clearly shows that the proposed MLR-PC has a superior performance on TMC2007 and Ohsumed datasets, while it has an acceptable result on the Genbase dataset, with respect to other MLL algorithms.

4.3 Discussion

The predictive performance of the PTMs varies according to dataset characteristics. Two types of multi-label datasets could be distinguished. The first type is the datasets with strong positive correlations

among labels, such as the Yeast, Emotions, Flags and TMC2007 datasets. For this type, the correlation-based PTMs show a superior performance, using most of the evaluation metrics. The second type is the datasets with weak positive correlations among labels due to low cardinality (Scene and Genbase) or because the dataset does not have significant positive high-order correlations among labels, like Ohsumed dataset. For this type, the correlation-based PTMs show either a quite limited improvement or no improvement at all when compared against the frequency-based PTMs.

Among the correlation-based PTMs, the HSDF shows the best predictive performance and HAPCF and HAPCSDF nearly show an equal predictive performance. Among the frequency-based PTMs, the LFL transformation method shows a better predictive performance than the MFL transformation method.

In general, as the total number of the captured positive correlations among labels increases, the predictive performance of the MLR-PC becomes better. The only exception for this finding is the LFL, in which the total number of the captured positive correlations is the highest when using the LFL as a transformation method. Nevertheless, the predictive performance of the MLR-PC algorithm is not affected greatly by this large number of positive correlations due to the limited exploitation of these positive correlations because of the small frequency of the labels that exploited these positive correlations. Table 9 shows the total number of the captured positive correlations among labels with respect to the PTM being used.

Table 9. Total number of the captured positive correlations among labels with respect to the PTM being used.

Dataset	**HAPCF**	**HSDF**	**HAPCSDF**	**MFL**	**LFL**
Yeast	10	16	11	1	19
Emotions	4	4	4	1	4
Flags	8	7	8	3	9
Genbase	14	16	15	2	18
TMC2007	3	3	3	0	3

5. CONCLUSIONS AND FUTURE WORK

In this paper, a new MLR algorithm called MLR-PC that captures positive correlations among class labels has been proposed. The captured positive correlations are exploited in the transformation step as well as in constructing a multi-label classifier. MLR-PC is a flexible algorithm, since any classifier could be used as a base classifier. Empirical analysis using different multi-label datasets, such as Yeast, Scene, Emotions and Flags, show that the MLR-PC algorithm is superior to other existing multi-label algorithms on several datasets, especially on those datasets with high cardinality.

High cardinality of a dataset is a strong evidence on the existence of significant correlations among labels and hence, MLR-PC algorithm showed a better performance with these datasets when utilizing correlation-based PTMs. Therefore, it is highly recommended to adopt a transformation criterion that considers the correlations among labels, especially with high-cardinality datasets, such as Yeast, Flags and TMC2007 datasets.

As for future work, more research that considers new PTMs based on the positive correlations among labels should be conducted. Also, capturing local and positive correlations among labels is a promising approach, especially in datasets with low cardinality, like the Genbase and Scene datasets.

REFERENCES

[1] R. Sousa and J. Gama, "Multi-label Classification from High-speed Data Streams with Adaptive Model Rules and Random Rules," Progress in Artificial Intelligence, vol. 7, pp. 1-11, 2018.

[2] C. Zeng, W. Zhou, T. Li, L. Shwartz and G. Y. Grabarnik, "Knowledge Guided Hierarchical Multi-label Classification over Ticket Data," IEEE Transactions on Network and Service Management, vol. 14, no. 2, pp. 246-260, 2017.

[3] E. Gibaja and S. Ventura, "A Tutorial on Multi-label Learning," ACM Computing Surveys, vol. 47, no. 3, pp. 1-39, 2015.

[4] J. Huang, X. Qu, G. Li, F. Qin, X. Zheng and Q. Huang, "Multi-view Multi-label Learning with View-label-specific Features," IEEE Access, vol. 7, pp. 100979-100992, 2019.

[5] S. Xu, X. Yang, H. Yu, D. J. Yu, J. Yang and E. C. Tsang, "Multi-label Learning with Label-specific Feature Reduction," Knowledge-based Systems, vol. 104, pp. 52-61, 2016.

[6] J. Huang, G. Li, S. Wang, Z. Xue and Q. Huang, "Multi-label Classification by Exploiting Local Positive and Negative Pairwise Label Correlation," Neurocomputing, vol. 257, pp. 164-174, 2017.

[7] S. J. Huang, Z. H, Zhou and Z. Zhou, "Multi-label Learning by Exploiting Label Correlations Locally," Proceedings of the 26th AAAI Conf. on Artificial Intelligence, pp. 949-955, Palo Alto, CA: AAAI, 2012.

[8] X. Kong, B. Cao and P. S. Yu, "Multi-label Classification by Mining Label and Instance Correlations from Heterogeneous Information Networks," Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 614-622, [Online], Available: https://doi.org/10.1145/2487575.2487577, 2013.

[9] J. Lee, H. Kim, N. R. Kim and J. H. Lee, "An Approach for Multi-label Classification by Directed Acyclic Graph with Label Correlation Maximization," Information Sciences, vol. 351, pp. 101-114, 2016.

[10] M. Alluwaici, A. K. Junoh, F. K. Ahmad, M. F. M. Mohsen and R. Alazaidah, "Open Research Directions for Multi-label Learning," Proc. of IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), pp. 125-128, Penang, Malaysia, 2018.

[11] R. E. Schapire and Y. Singer, "BoosTexter: A Boosting-based System for Text Categorization," Machine Learning, vol. 39, no. 2-3, pp. 135-168, 2000.

[12] R. AlShboul, F. Thabtah, N. Abdelhamid and M. Al-Diabat, "A Visualization Cybersecurity Method Based on Features' Dissimilarity," Computers & Security, vol. 77, pp. 289-303, 2018.

[13] M. R. Boutell, J. Luo, X. Shen and C. M. Brown, "Learning Multi-label Scene Classification," Pattern Recognition, vol. 37, no. 9, pp. 1757-1771, 2004.

[14] A. Dimou, G. Tsoumakas, V. Mezaris, I. Kompatsiaris and I. Vlahavas, "An Empirical Study of Multi-label Learning Methods for Video Annotation," Proceedings of the 7th IEEE International Workshop on Content-based Multimedia Indexing, pp. 19-24, New York, NY, USA, 2009.

[15] S. Peters, L. Denoyer and P. Gallinari, "Iterative Annotation of Multi-relational Social Networks," Proc. of the IEEE International Conference on Advances in Social Networks Analysis and Mining, pp. 96-103, DOI: 10.1109/ASONAM.2010.13, Odense, Denmark, 2010.

[16] S. H. Ma, H. B. Le, B. H. Jia, Z. X. Wang, Z. W. Xiao, X. L. Cheng et al., "Peripheral Pulmonary Nodules: Relationship between Multi-slice Spiral CT Perfusion Imaging and Tumor Angiogenesis and VEGF Expression," BMC Cancer, vol. 8, no. 1, DOI: 10.1186/1471-2407-8-186, 2008.

[17] K. Trohidis, G. Tsoumakas, G. Kalliris and I. P. Vlahavas, "Multi-label Classification of Music into Emotions," In: J. P. Bellow, E. Chew and D. Turnbull (Eds.), Proceedings of the 9th International Conf. on Music Informative Retrieval, pp. 325-330, Philadelphia, PN: Drexel University, USA, 2008.

[18] Z. Barutcuoglu, R. E. Schapire and O. G. Troyanskaya, "Hierarchical Multi-label Prediction of Gene Function," Bioinformatics, vol. 22, no. 7, pp. 830-836, 2006.

[19] A. Elisseeff and J. Weston, "A Kernel Method for Multi-labelled Classification," Advances in Neural Information Processing Systems, Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic (NIPS'01), pp. 681-387, 2001.

[20] A. Skabar, D. Wollersheim and T. Whitfort, "Multi-label Classification of Gene Function using MLPs," Proceedings of the 2006 IEEE International Joint Conference on Neural Network, pp. 2234-2240, DOI: 10.1109/IJCNN.2006.247019, Vancouver, BC, Canada, 2006.

[21] A. Chan and A. A. Freitas, "A New Ant Colony Algorithm for Multi-label Classification with Applications in Bioinfomatics," Proc. of the 8th Annual Conf. on Genetic and Evolutionary Computation (GECCO'06), pp. 27-34, [Online], Available: https://doi.org/10.1145/1143997.1144002, 2006.

[22] S. Diplaris, G. Tsoumakas, P. A. Mitkas and I. Vlahavas, "Protein Classification with Multiple Algorithms," Proc. of Panhellenic Conference on Informatics (PCI), pp. 448-456, Part of the LNCS, vol. 3746, Springer, Berlin, Heidelberg, 2005.

[23] K. Kawai and Y. Takahashi, "Identification of the Dual Action Antihypertensive Drugs Using TFS-based Support Vector Machines," Chem-Bio Informatics Journal, vol. 10, no. 9, pp. 41–51, Retrieved from: http://www.cbi.or.jp, 2010.

[24] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler and L. Schmidt-Thieme, "Multi-relational Matrix Factorization Using Bayesian Personalized Ranking for Social Network Data," Proceedings of the 5th ACM International Conference on Web Search and Data Mining, pp. 173-182, [Online], Available: https://doi.org/10.1145/2124295.2124317, 2012.

[25] L. Tang and H. Liu, "Relational Learning *via* Latent Social Dimensions," Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'09), pp. 817-826, [Online], Available: https://doi.org/10.1145/1557019.1557109, 2009.

[26] T. Li, C. Zhang and S. Zhu, "Empirical Studies on Multi-label Classification," Proc. of the 18th IEEE Int. Conf. on Tools with Artificial Intelligence (ICTAI'06), pp. 86-92, Arlington, VA, USA, 2006.

[27] O. A. Nassar and N. A. Al Saiyd, "The Integrating between Web Usage Mining and Data Mining Techniques," Proc. of the 5th IEEE International Conference on Computer Science and Information Technology, pp. 243-247, Amman, Jordan, 2013.

[28] J. Huang, Q. Feng, Z. Xiao, C. Zekai, Y. Zhixiang, Z. Weigang and H. Qingming, "Improving Multi-label Classification with Missing Labels by Learning Label-specific Features," Information Sciences, vol. 492, pp. 124-146, 2019.

[29] J. Read, B. Pfahringer, G. Holmes and E. Frank, "Classifier Chains for Multi-label Classification," Machine Learning, vol. 85, no. 3, p. 333, 2011.

[30] R. Alazaidah, F. Thabtah and Q. Al-Radaideh, "A Multi-label Classification Approach Based on Correlations among Labels," International Journal of Advanced Computer Science and Applications, vol. 6, no. 2, pp. 52-59, 2015.

[31] S. S. Ibrahiem, S. S. Ismail, K. A. Bahnasy and M. M. Aref, "Convolutional Neural Network Multi-emotion Classifiers," Jordanian Journal of Computers and Information Technology (JJCIT), vol. 5, no. 2, pp. 97-108, Aug. 2019.

[32] R. Alazaidah, F. K. Ahmad and M. F. M. Mohsin, "A Comparative Analysis between the Three Main Approaches that are Being Used to Solve the Problem of Multi-label Classification," International Journal of Soft Computing, vol. 12, no. 4, pp. 218-223, 2017.

[33] M. L. Zhang and Z. H. Zhou, "ML-KNN: A Lazy Learning Approach to Multi-label Learning," Pattern Recognition, vol. 40, no. 7, pp. 2038-2048, 2007.

[34] J. Fürnkranz and E. Hüllermeier, "Pairwise Preference Learning and Ranking," Proc. of the European Conference on Machine Learning (ECML), pp. 145-156, LNCS, vol. 2837, Springer, Heidelberg, 2003.

[35] J. Fürnkranz, E. Hüllermeier, E. L. Mencía and K. Brinker, "Multilabel Classification *via* Calibrated Label Ranking," Machine Learning, vol. 73, no. 2, pp. 133-153, 2008.

[36] G. Tsoumakas, I. Katakis and I. Vlahavas, "Effective and Efficient Multi-label Classification in Domains with Large Number of Labels," Proc. of ECML/PKDD 2008 Workshop on Mining Multi-dimensional Data (MMD'08), vol. 21, pp. 53-59, 2008.

[37] J. Read, "A Pruned Problem Transformation Method for Multi-label Classification," Proc. of 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008), pp. 143-150, 2008.

[38] G. Tsoumakas and I. Vlahavas, "Random k-labelsets: An Ensemble Method for Multi-label Classification," Proc. of the European Conference on Machine Learning, pp. 406-417, LNCS, vol. 4701, Springer, Berlin, Heidelberg, 2007.

[39] M. Zhang and L. Wu, "Lift: Multi-label Learning with Label-specific Features," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 37, no. 1, pp. 107-120, DOI: 10.1109/TPAMI.2014.2339815. 2015.

[40] L. Rokach, A. Schclar and E. Itach, "Ensemble Methods for Multi-label Classification," Expert Systems with Applications, vol. 41, no. 16, pp. 7507-7523, 2014.

[41] R. Alazaidah, F. K. Ahmad, M. F. M. Mohsen and A. K. Junoh, "Evaluating Conditional and Unconditional Correlations Capturing Strategies in Multi-label Classification," Journal of Telecommunication, Electronic and Computer Engineering (JTEC), vol. 10, no. 2-4, pp. 47-51, 2018.

[42] E. Frank and H. Witten, "Generating Accurate Rule Sets without Global Optimization," Proceedings of the 15th International Conference on Machine Learning (ICML '98), pp. 144–151, 1998.

[43] T. Scheffer, "Finding Association Rules that Trade Support Optimally Against Confidence," Intelligent Data Analysis, vol. 9, no. 4, pp. 381-395, 2005.

[44] R. R. Bouckaert, "Properties of Bayesian Belief Network Learning Algorithms," Uncertainty Proceedings

1994, pp. 102-109, Morgan Kaufmann, [Online], Available: https://doi.org/10.1016/B978-1-55860-332-5.50018-3, 1994.

[45] W. W. Cohen, "Fast Effective Rule Induction," Proceedings of the 12th International Conference on Machine Learning, pp. 115-123, Tahoe City, California, 1995.

[46] M. Sumner, E. Frank and M. Hall, "Speeding up Logistic Model Tree Induction," Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases Discovery (PKDD'05), pp. 675-683, Heidelberg: Springer-Verlag, DOI: 10.1007/11564126_72, 2005.

[47] R. Kohavi, "The Power of Decision Tables" Proceedings of the 8th European Conference on Machina Learning (ECML'95), pp. 174-189, Berlin, Heidelberg: Springer-Verlag, DOI: 10.1007/3-540-59286-5_57, 1995.

[48] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA Data Mining Software: An Update, " ACM SIGKDD Explorations Newsletter, vol. 11, no.1, pp. 10-18, 2009.

[49] G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek and I. Vlahavas, "Mulan: A Java Library for Multi-label Learning," Journal of Machine Learning Research, vol. 12, no. Jul., pp. 2411-2414, 2011.

[50] S. Zhu, X. Ji, W. Xu and Y. Gong, "Multi-labelled Classification Using Maximum Entropy Method," Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05), pp. 274-281, [Online], available: https://doi.org/10.1145/1076034.

1076082 2005.

[51] R. Alazaidah, F. Ahmad and M. Mohsin, "Multi-label Ranking Based on Positive Pairwise Correlations among Labels," The International Arab Journal of Information Technology, vol. 17, no. 4, DOI: 10.34028/iajit/17/4/2, 2019.

[52] J. Huang, Z. Pingzhao, Z. Huiyi, L. Guorong and R. Haowei, "Multi-label Learning *via* Feature and Label Space Dimension Reduction," IEEE Access, vol. 8, pp. 20289-20303, 2020.

[53] F. Alshraiedeh, S. Hanna and R. Alazaidah, "An Approach to Extend WSDL-based Data Types Specification to Enhance Web Services Understandability," International Journal of Advanced Computer Science and Applications, vol. 6, no. 3, pp. 88-98, 2015.

ملخص البحث:

يعدّ تصنيف الأوسام المتعددة نوعاً عاماً من أنواع التصنيف والذي جذب اهتمام العديد من الباحثين في العقدين الماضيين، نظراً لقابليته للتطبيق في الكثير من المجالات الحديثة؛ مثل تصنيف المشاهد، وتكنولوجيا المعلومات الحيوية، وتصنيف النصوص، من بين مجالاتٍ أخرى. ويسمح هذا النوع من التصنيف بأن تكون الأمثلة مرتبطة بأكثر من وَسْم واحدٍ من أوسام الأصناف في آنٍ واحد. إنّ ترتيب أوسام الأصناف مسألة حاسمة في البحث المتعلق بتصنيف الأوسام المتعددة؛ نظراً لما له من تأثير مباشر في أداء المصنِّفات النهائية، لا سيما أنّ الأوسام التي تحتل المراتب العليا تحصل على فرصةٍ أعلى للتطبيق. تقدم هذه الورقة خوارزمية جديدة لترتيب الأوسام المتعددة، تسمى "ترتيب الأوسام المتعددة بناءً على الارتباطات الموجبة بين الأصناف (MLR-PC)". وتلتقط الخوارزمية المقترحة الارتباطات الموجبة بين الأوسام للتقليل من الحيّز الضخم للبحث وتحدد المرتبة الحقيقية لكل وسمٍ من أوسام الأصناف في المسائل المتعلقة بتصنيف الأوسام المتعددة.

والأهم من ذلك أنّ الخوارزمية المقترحة تستخدم طرقاً مبتكرة لتحويل المسائل، الأمر الذي يسهل استغلال الارتباطات الموجبة الدقيقة بين الأوسام. وهذا من شأنه أن يحسن الأداء التنبؤي لنماذج التصنيف المشتقة. وقد كشفت النتائج التجريبية باستخدام مجموعة بيانات متعددة الأوسام وخمسة من مقاييس التقييم أن الخوارزمية المقترحة (MLR-PC) تتفوق على الخوارزميات الأخرى شائعة الاستخدام في عمليات التصنيف.