Multi Label Ranking Based on Positive Pairwise Correlations Among Labels

Article *in* International Arab Journal of Information Technology · July 2019

DOI: 10.34028/iajit/17/4/2

Raed Alazaidah
Universiti Utara Malaysia

Farzana Kabir Ahmad
Universiti Utara Malaysia

Multi Label Ranking Based on Positive Pairwise Correlations Among Labels

Raed Alazaidah, Farzana Ahmad, and Mohamad Mohsin
School of Computing, Universiti Utara Malaysia, Malaysia

Abstract: *Multi-Label Classification (MLC) is a general type of classification that has attracted many researchers in the last few years. Two common approaches are being used to solve the problem of MLC: Problem Transformation Methods (PTMs) and Algorithm Adaptation Methods (AAMs). This Paper is more interested in the first approach; since it is more general and applicable to any domain. In specific, this paper aims to meet two objectives. The first objective is to propose a new multi-label ranking algorithm based on the positive pairwise correlations among labels, while the second objective aims to propose new simple PTMs that are based on labels correlations, and not based on labels frequency as in conventional PTMs. Experiments showed that the proposed algorithm overcomes the existing methods and algorithms on all evaluation metrics that have been used in the experiments. Also, the proposed PTMs show a superior performance when compared with the existing PTMs.*

Keywords: *Correlations among labels; multi-label classification; multi-label ranking; problem transformation methods.*

Received August 23, 2017; accepted August 2, 2018

1. Introduction

Classification is one of the most interested task in data mining that has attracted many researchers and scholars in the last few decades [5]. In general, classification could be categorized into two main types: Single Label Classification (SLC) and Multi-Label Classification (MLC). In SLC, each instance is associated with only one class label from a set of disjoint class labels. If the total number of classes in the data set equals two, then the problem is called binary classification, otherwise, the problem is called multi class classification. On the other hand, MLC allows instances in the data set to be associated with one or more class labels at the same time [2, 3].

Two main differences between SLC and MLC. In SLC, labels are mutual exclusive, while they are not in MLC, and they do have correlations and dependencies among them. The second main difference is that the problem search space in SLC is quite limited when comparing with MLC. For example, if the total number of labels in SLC is 20, then, the problem search space consists of only 20 possible labels. While if the problem is a MLC, and the total number of labels equals 20, then, the problem search space consists of 2^{20} possible labels combinations, which is more than one million possible combinations. Thus, the problem of MLC is very complicated when compared to the problem of SLC; due to the existing dependencies and correlations among labels, which cause a huge problem search space [3].

According to Gibaja and Ventura [7], Multi-Label Learning (MLL) includes two different tasks: MLC and Label Ranking (LR). While the goal in MLC is to divide labels into two groups (relevant labels, irrelevant labels), based on a specific function for any test instance, the goal of LR is to order all labels, according to the relevance of the labels to a given test instance. Multi-Label Ranking (MLR) is a generalization of both tasks of learning, which outputs a bipartition and a ranking at the same time.

Two main common approaches are being used to solve the problem of MLC. The first approach called Problem Transformation Methods (PTMs), while the second approach called Algorithm Adaptation Methods (AAMs). The former transforms the problem of MLC into one or several SLC problems, and then, trains a single label classifier or more on the transformed datasets. The predictions of these single label classifiers are merged to form a multi-label predictions. AAMs adapts a single label classification algorithm to handle multi-label data. According to Read, Pfahringer, Holmes, and Frank [12], PTMs are preferable over AAMs; since they are more simpler, more general, and not a domain specific like AAMs.

Recently, there is a strong belief among many researchers regarding the importance of capturing the correlations and dependencies among labels, in order to reduce the huge search space of the MLC problem, as well as to improve the predictive performance of a MLC classifier [1, 4]. Hence, this paper is much more interested in capturing positive pairwise correlations among labels, as well as maximizing the exploitation of these discovered correlations. Thus, the discovered correlations will be exploited in several different steps of the proposed algorithm such as the transformation step, multi-label classifier construction step and the ranking of the relevant labels step.

In this paper, the researchers are more interested in the approach of PTMs to handle MLC; because we have a firm belief that this approach is more simpler, more general and applicable to any domain, and the AAMs approach in its essence, explicitly or implicitly performs a transformation step [12]. Hence, the outputs of this research will be applicable and generalized to both approaches.

Since the transformation step is one of the main steps in many PTMs as well as AAMs, it is very important to think out of the box regarding this step. In Fact, most methods and algorithms utilize this step in only one task, which is the transformation of the multi-label dataset into single label dataset. Nevertheless, we believe this step should be considered with more attention. Thus, we are proposing to use new PTMs based on the positive pairwise correlations among labels, and not based on labels frequency as in the conventional PTMs. Transforming multi-label dataset into a single label dataset using the correlations among labels as a transformation criterion will facilitate the utilization and exploitation of the significant correlations among labels greatly, and consequently improve the final predictive performance of the multi-label classifier.

Generally speaking, capturing the correlations among labels may increase the complexity of any proposed multi-label algorithm, as well as the running time. Thus, this task should be considered wisely and must be justified correctly. Hence, the proposed multi-label ranking algorithm captures significant positive pairwise correlations among labels, and exploits it in three different main step (transformation step, multi-label classifier construction step and the ranking step). We believe, the cost paid for capturing and exploiting the correlations among labels will be justified by the fair enhancement of the final multi-label classifier predictive performance.

This paper is organized as following. The next section briefly surveys the literature review. Section 3 introduces the proposed PTMs and the proposed MLR algorithm. Section 4 concludes and describes few research works.

2. Related Work

Many algorithms have been proposed to solve the problem of MLC. These algorithms could be categorized based on the degree of correlations among labels, that has been considered in the learning step, into three approaches [16]. The first approach is known as a first order approach, and tackles the problem of MLC by considering labels to be mutual exclusive, and never considers the correlations among labels.

One of the most popular algorithms that follows the approach of first order is the Binary Relevance (BR) algorithm. This algorithm assumes labels in a multi-label dataset to be independent, and ignores any possible correlations and dependencies among them. It is similar to the concept of One-versus-All (OVA) that has been extensively used in multi class classification [4]. BR divides the original multi-label dataset into (k) single label datasets, where each dataset is specific to one label only. Instances in this dataset are either labelled as "True", if the original instance is associated with this label, or labelled as "False" otherwise. A binary classifier then is trained to predict the relevance of a test instance to each label.

BR has several advantages over other methods such as: its simplicity in handling the problem of MLC, the ability to rum the algorithm in parallel, and therefore, speedup the process of constructing a multi-label classifier, the low possibility of overfitting, and finally, BR has a low computational complexity when compared with other methods [12].

Although BR is a simple method that inspired many researchers, but it has been criticized for several drawbacks like assuming labels to be independent, and not taking labels correlations into account. Another drawback is the huge loss of information regarding to ignoring labels correlations and dependencies. A third drawback appears when there are many labels in the dataset, which complicated the training phase of BR [14].

The simplicity of BR has inspired many scholars to design new methods that try to overcome the disadvantages of BR. One of the first method that was inspired and designed based on BR is the Classifier Chains (CC) algorithm [12]. Similar to BR, CC algorithm divides a multi-label dataset into (k) different single label datasets, then it trains a binary classifier for each label. A chain of classifiers then is build, where binary attributes are added to each classifier for all of the predictions of the previous classifiers.

Considering the prediction of the previous classifiers in the chain is the CC's way to overcome BR drawback of not taking labels correlations into account. In average (k/2) binary attributes are added to every instance, but this addition has a small impact on the computational complexity of CC, which is almost close to BR computational complexity [7]. Execution of CC could be done in parallel like BR in the training phase, while it has to be serialized in the testing phase. The main drawback of CC is in determining the optimal order of the chains, where it has been proved, and stated by its author himself that different orders of the chains give different predictions, and therefore, will affect the accuracy of the final result [12].

To solve the problem of different orders give different predictions, a new version of CC was presented by the same author, using an Ensemble of Classifier Chains (ECC) which used a random ordering of the chains, in addition to random training subsets [12].

In Goncalves, Plastino, and Freitas [8], an attempt to optimize and determine the best order of the chains was presented. The authors proposed to use the capability of the Genetic Algorithm (GA) in order to determine the optimal chains' order. The proposed algorithm Genetic Algorithm for optimizing the label ordering in multi-label Classifier Chain (GACC) needs to be evaluated on more datasets using several evaluation measures that are more related to MLC.

Another algorithm that follows the approach of the first order is the Multi Label K Nearest Neighbour (ML-KNN) [17]. ML-KNN is one of the first algorithm that adopted lazy learning to handle MLC. ML-KNN is based on the popular algorithm KNN. The first step in ML-KNN is to determine the (k) nearest neighbours for each new instance in the training test. The Maximum A Posteriori (MAP) is utilized to determine the label set of the test instance. MAP is utilized based on the statistical information of the label sets of the neighbouring instances.

The second approach is known as a second order approach, and takes into consideration only pairwise correlations among labels. Two main popular methods that follow the second order approach. The first method is called Ranking by Pairwise Comparisons (RPC) [6]. 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 the both labels [12]. 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 the high quadratic complexity that makes it a very bad choice when dealing with large number of labels. Another limitation is that RPC consumes a large space of the main memory to construct (k (k-1) / 2) datasets. The last limitation of RPC is that it does not have a split point between relevant and irrelevant labels [7].

To overcome the last drawback of RPC, the Calibrated Label Ranking (CLR) was proposed. CLR 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 [6]. As in RPC, the CLR method suffers from space complexity, and computational complexity also.

Back propagation for Multilabel Learning (BP-MLL) algorithm [17] is an adaptation of the traditional multi layer, feed-forward neural network to multi-label data. The net was trained with 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 networks methods.

The third approach that is known as a high order approach considers a high order of correlations among all labels in the labels set, or among a subset of labels [16]. One of the high order approach algorithm that captures and exploits high order correlations among labels is the Label Powerset (LP) algorithm. LP considers each unique set of labels combinations in the training set as a new class label, in a multi class classification problem. LP handles the problem of MLC by transforming it into a multi class classification problem [3]. LP has the ability to exploit the correlations among labels in a simple and effective way, but it suffers from several limitations. Firstly, LP is able to predict only those classes (labels combinations) that exist in the training set, and cannot predict new label sets that appear only in the test set. This is a major limitation, since it will lead to an overfitting problem. Secondly, there is a big possibility to suffer from the problem of imbalance class distribution when using LP and the possibility increases as the number of labels and the number of distinct label sets increase [15]. Thirdly, the computational complexity of LP is exponential with respect to the number of labels, and the number of distinct label sets. Its complexity is upper bounded by min (m, 2^k), but usually it is less than that. LP works well with small datasets, but quickly fails with moderate and large datasets.

As in BR, LP has inspired many researchers to design new methods that consider labels correlations into account. The first method that was based on LP, and tried to enhanced LP by overcoming the problem of imbalance class distribution was the Pruned Set (PS) method [11]. PS prunes all the label sets that have a frequency less than a specific user defined threshold. This strategy may solve the problem of high computational complexity of LP, and the problem of imbalance class distribution, but at the same time it imposes a new problem, which is the information loss due to the pruned labels combinations. The author proposed to use small subsets of the pruned combinations, that are frequent in the dataset in the final prediction of the classifier.

The Ensemble of Pruned Set (EPS) method constructs a number of pruned sets through sampling the training set, and builds the final prediction using voting schema, and a user predefined threshold, in order to form new combinations of labels [3].

Tsoumakas and Vlahavas [13] proposed their famous algorithm RAndom k labEL set (RAkEL) based on LP method. RAkEL constructs an ensemble of LP classifiers, where each classifier is trained using a smaller random subset that consists of (k) labels. To classify a new instance. The outputs of all classifiers are averaged per label with respect to a predefined threshold. RAkEL is a problem transformation method that is algorithm independent, thus any single label classifier could be used with it. The authors of RAkEL recommended to use either C4.5 [10] or Support Vector Machine (SVM) [9] as a base classifier for RAkEL. This recommendation was based on an intensive experiment conducted by the authors themselves to determine the best base classifier for their algorithm.

RAkEL manages to consider labels correlations into account, and at the same time avoids the computational complexity of LP, by considering smaller subsets of labels combinations. RAkEL has the ability to predict label sets that did not appear in the training phase, which LP failed to do. RAkEL is less possible to suffer from the overfitting or imbalance class distribution, due to minimizing the size of the label sets [1].

Regardless of the great advantages and enhancements of RAkEL over LP, but it still suffers from several limitations such as: the huge loss of information that is ignored to solve the problem of high computational complexity, and the imbalance class distribution. Another limitation is in determining the optimal value of (k). The authors suggested the best value of (k) to be 3, but they never showed why and how they reached to this value [15]. The authors never discussed the ability to determine the value of (k) automatically, and what is the relation between labels cardinality and the value of (k), and whether RAkEL has the ability to handle multi-label datasets with large label cardinality [2].

To summarize, the first approach may have the advantage of being simple and easy to implement, but suffers from a major limitation of not taking labels correlations into account. Hence, ignoring the basic principle of MLC problem, that is, the existence of dependencies and correlations among labels [2].

Although, the second order approach manages to enhance the predictive performance on some multi-label datasets, but it has a limitation of addressing the correlations among labels into a certain extent. Therefore, it seems to be relatively effective, especially in datasets with large number of labels, where it needs to perform $\binom{q}{2}$ pairwise comparisons [4].

For the high order approach, it can be clearly seen that, it makes a better enhancement in the predictive performance, especially in moderate and large size datasets. Nevertheless, this approach has significant limitations in complexity, and tends to be more computationally demanding and time consuming [4].

3. Multi Label Ranking based on Positive Pairwise Correlations among Labels (MLR-PPC)

According to Read *et al.* [12], the first step in most PTMs is the transformation step that aims to fit the data into any single label classifier. All existing PTMs depend on label's frequency as a transformation criterion. Thus, any multi-label dataset is transformed to be associated with the Most Frequent Label (MFL) that is associated with it or the Least Frequent Label (LFL). The ignore transformation method discards any multi-label instance, and considers only instances that are associated with one label only.

The previous PTMs share several limitations such as the huge loss of information; due to the transformation step, imbalance class distribution especially when choosing the MFL as a transformation method, and not facilitating the step of capturing and exploiting the correlations among labels.

Thus, in this paper, we claim that the transformation step should be considered wisely. The transformation step should not only utilized to transform the multi-label dataset into a single label dataset, but also should facilitate capturing the correlations among labels. Hence, the proposed PTMs consider the positive pairwise correlations among labels, and not the label's frequency as in the existing PTMs.

To transform a multi-label dataset into single label dataset using correlations among labels as a transformation criterion, we need to capture these correlations first. Thus, the label space of the multi-label dataset is extracted firstly, and then an adapted version of Predictive Apriori is applied on the extracted label space, to discover the positive pairwise correlations only, in the form of (IF L1=1 THEN L2=1). After discovering all the positive correlations for each label in the label set, the labels are ordered according to their highest positive pairwise correlations or their highest Standard Deviation between the discovered positive correlations for each label, or according to the summation of the highest accurate positive correlations accuracy and the Standard Deviation of the that label. The three proposed PTMs are described next.

a. Highest Accurate Positive Correlations First (HAPCF)

1. Discover all positive pairwise association rules in the label space of the dataset, using the adapted version of Predictive Apriori.
2. For every label in the label set, identify the maximum accurate positive association rule with other labels.
3. Order the labels according to the maximum accurate positive association rule in a descending way.
4. Use the order discovered in the previous step to transform the multi-label dataset into a single label dataset.

b. Highest Standard Deviation First (HSDF)

1. Discover all positive pairwise association rules in the label space of the dataset, using the adapted version of Predictive Apriori.
2. For every label in the label set, calculate the Standard Deviation among all the discovered positive association rules.
3. Order the labels according to the Standard Deviation calculated in the previous step in a descending way.

4. Use the order discovered in the previous step to transform the multi-label dataset into a single label dataset.

c. Highest Accurate Positive Correlations and Standard Deviation First (HAPCSDF)

1. Discover all positive pairwise association rules in the label space of the dataset, using the adapted version of Predictive Apriori.
2. For every label in the label set, compute the summation of the maximum accurate positive association rule and the Standard Deviation among all the discovered positive rules for that label.
3. Order the labels according to the summation computed in the previous step in a descending way.
4. Use the order discovered in the previous step to transform the multi-label dataset into a single label dataset.

The three proposed PTMs will be used as transformation methods, and applied to four multi-label datasets with another two existing PTMs (MFL, LFL); to prove the effectiveness of the proposed PTMs. The evaluation process will not only considered the evaluation of the base classifier using the five PTMs, but also will consider the final accuracy of the proposed MLR-PPC, when applied using the proposed and the existing transformation method, since the proposed PTMs are expected to improve the predictive performance of a multi-label classifier by facilitating capturing of significant positive correlations among labels. Hence, increase the accuracy of the final multi-label classifier.

In order to maximize the exploitation of the discovered positive correlations among labels, and justify the additional step of capturing these correlations, the discovered positive pairwise correlations will be exploited in another two important steps. The first step is the step of constructing a multi-label classifier, while the second step is the ranking step. Algorithm 1 depicts the proposed Multi Label ranking algorithm based on Positive Pairwise Correlations among labels (MLR-PPC).

MLR-PPC starts with extracting the label space of the input multi-label dataset, and considers it as a transactional dataset. The second step in MLR-PPC is to apply an adapted version of Predictive Apriori algorithm, where an additional filtering step has been added to Predictive Apriori algorithm to consider only positive association rules in the form of (IF L1=1 THEN L2=1), where L1 and L2 are two disjoint labels. The third step is the transformation step of the input multi-label dataset into a single label dataset. After transforming the input multi-label dataset into a single label dataset, MLR-PPC uses any rule-based single label classifier on the transformed data set to construct a single label classifier. The fifth step in MLR-PPC aims to convert the single label classifier constructed earlier to a multi-label classifier. To achieve this goal, MLR-PPC modifies the consequent of every rule's consequent in the single label classifier, by amending the best (n) pairwise correlations for the label that exists in the consequent of the classification rule under processing. The value of (n) equals to the Label Cardinality (LC) -1, where LC refers to the average number of labels per instance. The last step in MLR-PPC is the evaluation step, where the new multi-label classifier is tested against new data.

Algorithm 1: MLR-PPC Algorithm

```
Input: Multi-label dataset (D), minacc threshold
Output: Multi-label classifier

TD=Label Space (D)
For each x in TD
 {
    Generate all Positive Pairwise Association Rules (PARs) in a
      form of <<x    y→>>, where y has a lower transformation
      order than x, using Predictive Apriori algorithm.
}
SLD= Transform (D, {HAPCF, HSDF, HAPCSDF}) //SLD: the
transformed Single Label Dataset.
Classify (SLD)      //Using PART Algorithm
For each rule (r) generated by PART
{
    Modify the consequent thus new consequent = consequent +
    [Z], where [Z] = labels with the best positive pairwise
    correlations with the label in the consequent of r.
}
Use the new multi-label classifier to test the data
```

4. Evaluation of the Proposed PTMs and the Proposed MLR-PPC Algorithm

In this section, a description of the evaluation step of both the proposed PTMs, and the proposed MLR-PPC algorithm is presented. Firstly, a description of the proposed PTMs evaluation is introduced, then the evaluation of the proposed MLR-PPC is presented. It is very crucial to mention that the evaluation of the proposed PTMs is based on the accuracy of the single label classifier constructed using any of the proposed PTMs, and the final accuracy of the multi-label classifier when combined with any of the proposed PTMs, as well as the existing transformation methods.

4.1. Evaluation of the Proposed PTMs

The evaluation of the proposed PTMs consists of two phases. The first phase considers the accuracy of the base classifier trained on a transformed dataset using one of the existing (MFL, LFL) or the proposed PTMs (HAPCF, HSDF, and HAPCSDF). This phase is not significant as the second phase, since it considers only one class labels, while the second phase of evaluating the proposed PTMs is more significant, as it considers predicting several class labels and not only one class labels. The significance of the second phase becomes more and more important as LC of the dataset gets higher. In general, the final accuracy of any multi-label classifier depends on the accuracy of two tasks. The first task is the classification of the transformed dataset, while the second task is the predicting of all labels that have been discarded due to the transformation step. The second task has more affect on the final accuracy of the multi-label classifier as it considers more labels than the first task. For example, if the LC of a dataset is 4, then after transforming this multi-label dataset into single label dataset, 3 labels will be discarded in average per instance. Thus, it is very crucial that the proposed PTMs facilitate the task of predicting these discarded labels.

4.1.1. Evaluating the Proposed PTMs based on the Accuracy of the Base Classifiers

Four data set have been used in this research (Emotions, Flags, Yeast). Table 1 describes the main characteristics of each data set.

Table 1. Data sets main characteristics.

Dataset	Instances	Attributes	Labels	LC	Domain
Yeast	2417	103	14	4.327	Biology
Emotions	593	72	6	1.868	Media
Flags	194	19	7	3.392	Image
Scene	2712	294	6	1.074	Image

Each dataset has been transformed into a single label dataset 5 times, where each time a transformation method is used. The considered transformation methods are: MFL, LFL, HAPCF, HSDF, and finally HAPCSDF. Then, five different base classifiers (PART, ONER, Ridor, CR, JRIP) were trained on the transformed versions of the datasets. Table 2 depicts the accuracy of the five base classifiers on the transformed version of Emotions dataset.

Table 2. Evaluating the base classifiers on the transformed versions of Emotions dataset.

PTMs	PART	ONER	Ridor	CR	JRIP	Average
HAPCF	96.03	52.47	68.81	48.51	72.27	67.618
HSDF	96.03	52.47	68.81	48.51	72.27	67.618
HAPCSDF	96.03	52.47	68.81	48.51	72.27	67.618
LFL	96.03	52.47	68.81	48.51	72.27	67.618
MFL	**96.53**	**64.85**	**84.15**	**62.37**	**82.67**	**78.114**

Table 3 shows the accuracy of the five base classifiers when applied to the Flags dataset after transformation.

Table 3. Evaluating the base classifiers on the transformed versions of Flags dataset.

PTMS	PART	ONER	Ridor	CR	JRIP	Average
HAPCF	**92.3**	58.64	**76.92**	55.38	70.76	70.8
HSDF	89.23	64.61	66.15	56.92	60	67.382
HAPCSDF	89.230	63.076	75.384	50.7692	72.3077	70.15384
LFL	84.61	43.07	66.15	41.53	46.15	56.302
MFL	89.23	**75.38**	75.38	**75.38**	**81.53**	**79.38**

Table 4 shows the accuracy of the five base classifiers when applied to the Yeast dataset after transformation.

Table 4. Evaluating the base classifiers on the transformed versions of Yeast dataset.

PTMs	PART	ONER	Ridor	CR	JRIP	Average
HAPCF	**97.341**	63.8796	61.8729	60.5351	70.5686	70.83944
HSDF	92.307	43.1438	70.5686	26.087	48.8294	56.1873
HAPCSDF	93.624	61.4094	**76.5101**	57.3826	70.8054	71.94634
LFL	90.103	36.4548	63.5452	28.4281	45.4849	52.8032
MFL	95.65	**74.247**	74.247	**74.247**	**80.936**	**79.86624**

It can be clearly seen from the previous 3 tables that the MFL has the best accuracy over the 5 base classifiers on the three multi-label datasets. The proposed PTMs overcome the LFL on the three datasets using Accuracy as evaluation metric.

Table 5 to Table 7 summarize the evaluation of the proposed PTMs using four evaluation metrics (Precision, Recall, F1-Measure, and Receiver Operating Characteristic (ROC)) on the three datasets, averaged using the 5 base classifiers used previously.

Table 5. Evaluating the proposed PTMs using four evaluation metrics on Emotions dataset.

PTM	Precision	Recall	F1-Measure	ROC
HAPCF	0.6452	0.6762	0.6422	0.8072
HSDF	0.6452	0.6762	0.6422	0.8072
HAPCSDF	0.6452	0.6452	0.6452	0.6452
LFL	0.6452	0.6452	0.6452	0.6452
MFL	**0.7224**	**0.7814**	**0.7378**	**0.8224**

Table 6. Evaluating the proposed PTMs using four evaluation metrics on Flags dataset.

PTM	Precision	Recall	F1-Measure	ROC
HAPCF	0.6386	0.7078	0.663	**0.7804**
HSDF	0.6374	0.6738	0.6436	0.7614
HAPCSDF	0.6204	0.7016	0.648	0.7606
LFL	0.4616	0.5632	0.4822	0.74
MFL	**0.6722**	**0.7938**	**0.7162**	0.61

Table 7. Evaluating the proposed PTMs using four evaluation metrics on Yeast dataset.

PTM	Precision	Recall	F1-Measure	ROC
HAPCF	0.6132	0.703	0.6338	0.6804
HSDF	0.5346	0.5618	0.5148	**0.7358**
HAPCSDF	0.6266	0.7194	0.6554	0.7346
LFL	0.4746	0.5324	0.4762	0.7374
MFL	**0.7236**	**0.7938**	**0.7384**	0.7094

In general, and based on the evaluation metrics from the three previous tables, it can be seen clearly that the MFL transformation method overcomes all other transformation methods on all the three datasets. Also, the proposed PTMs overcome the LFL transformation method in mostly all cases for the three datasets.

To conclude, the evaluation process of the proposed PTMs based on the accuracy of the base classifier showed that the MFL is the best transformation method, while the LFL is the worst choice. For the proposed PTMs, it is clear that they showed an accepted performance when considering the first step of constructing the multi-label classifier. The following subsection describes the evaluation of the proposed

PTMs based on the final accuracy of the constructed multi-label classifier using MLR-PPC algorithm.

4.1.2. Evaluating the Proposed PTMs based on Facilitating Correlations Capturing

To evaluate the proposed PTMs based on facilitating the correlations capturing task, the proposed MLR-PPC algorithm was applied on the three multi-label datasets using the proposed PTMs (HAPCF, HSDF, and HAPCSDF), as well as the existing PTMs (MFL, LFL). PART algorithm have been used as a base classifier for the proposed MLR-PPC algorithm. Table 8 to Table 10 show the final Accuracy of the MLR-PPC algorithm, when applied using PART algorithm as a base classifier, and one of the PTMs from both existing (MFL, LFL) and proposed (HAPCF, HSDF, and HAPCSDF) transformation methods.

Table 8. Evaluating MLR-PPC using Emotions dataset.

MLR-PPC + PTMs	Accuracy
MLR-PPC-HAPCF	77.01
MLR-PPC-HSDF	77.01
MLR-PPC-HAPCSDF	77.01
MLR-PPC-LFL	76.53
MLR-PPC-MFL	61.29

Table 9. Evaluating MLR-PPC using Flags dataset.

MLR-PPC + PTMs	Accuracy
MLR-PPC-HAPCF	67.93
MLR-PPC-HSDF	61.83
MLR-PPC-HAPCSDF	60.32
MLR-PPP-LFL	60.78
MLR-PPC-MFL	50.32

Table 10. Evaluating MLR-PPC using Yeast dataset.

MLR-PPC + PTMs	Accuracy
MLR-PPC-HAPCF	55.67
MLR-PPC-HSDF	57.41
MLR-PPC-HAPCSDF	53.22
MLR-PPC-LFL	53.16
MLR-PPC-MFL	42.56

The three previous Tables show clearly that the evaluation of the proposed MLR-PPC algorithm using the proposed PTMs overcomes the case when using the existing PTMs.

4.2. Evaluation of the Proposed MLR-PPC Algorithm

Table 11 depicts a comparison between the proposed MLR-PPC algorithm and other multi-label learning algorithms using four multi-label datasets with different characteristics. The compared algorithms have been chosen to represent the three types of correlations capturing approaches (first order, second order, and high order).

Also, the chosen algorithms belong to both PTMs and AAMs approaches. Three multi-label evaluation metrics that are related to the classification task have been used to evaluate the proposed MLR-PPC algorithm (Accuracy (Acc), Exact Match, and Hamming Loss (H.L)).

Table 11. Evaluating the proposed MLR-PPC with respect to the existing algorithms.

Algorithm	Flags			Emotions			Yeast			Scene		
	Acc	Exact Match	H.L	Acc	Exact Match	H.L	Acc	Exact Match	H.L	Acc	Exact Match	H.L
MLR-PPC-HAPCF	67.93	24.60	17.30	77.01	60.80	09.70	55.67	18.70	14.52	0.908	0.863	0.0014
MLR-PPC-HSDF	61.83	2[illegible].00	18.90	77.01	60.80	09.70	57.41	19.00	13.11	0.908	0.863	0.0014
MLR-PPC-HAPCSDF	60.32	24.60	19.50	77.01	60.80	09.70	53.22	18.00	14.57	0.908	0.863	0.0014
MLR-PPC-LFL	60.78	30.70	17.50	76.53	60.30	10.30	53.16	15.70	14.58	0.881	0.858	0.0014
MLR-PPC-MFL	50.32	20.00	25.70	61.29	33.10	14.70	42.56	09.70	18.93	0.885	0.859	0.0014
BR	[illegible]7.63	07.69	27.47	55.10	30.70	18.80	55.20	20.10	19.30	0.64	0.617	0.009
LP	NG	NG	NG	58.40	35.10	19.80	52.30	26.00	20.60	0.7350	0.6960	0.0900
RAKEL	NG	NG	NG	59.20	34.10	18.60	49.30	16.30	20.70	0.6940	0.6620	0.0950
CC	55.87	20.00	29.89	58.40	34.90	19.70	52.10	25.40	21.10	0.7360	0.6690	0.1000
PS	NG	NG	NG	59.90	36.70	19.20	53.30	25.80	20.50	0.7510	0.7170	0.0840
ECC	56.00	19.10	28.80	28.20	00.07	49.40	29.90	24.30	46.20	0.2700	0.0070	0.4700
EPS	NG	NG	NG	59.90	36.60	19.30	53.70	25.30	20.70	0.7510	0.7150	0.0850
ML-KNN	55.50	09.80	28.40	36.60	14.30	26.20	52.00	18.90	19.30	0.69	0.643	0.085
BP-MLL	NG	NG	NG	27.60	27.60	43.30	18.50	18.50	32.20	0.21	0.212	0.057

Table 12 depicts a comparison between the proposed MLR-PPC algorithm and other multi-label learning algorithms on the four considered multi-label datasets and using two metrics that are most related to MLR task. The first metric is called One Error (1Error), and the second metric is called Coverage (Cov.).

Table 12. Evaluating the proposed MLR-PPC with respect to the existing algorithms.

	Flags		Emotions		Yeast		Scene	
	1Error	**Cov.**	**1Error**	**Cov.**	**1Error**	**Cov.**	**1Error**	**Cov.**
MLR-PPC-HAPCF	0.0769	2.8923	0.0390	1.1930	0.0530	5.1725	0.0550	0.13377
MLR-PPC-HSDF	0.1076	2.9384	0.0390	1.1930	0.0760	4.5975	0.0550	0.13377
MLR-PPC-HAPCSDF	0.1076	2.8307	0.0390	1.1930	0.0630	5.1075	0.0550	0.13377
MLR-PPC-LFL	0.1538	3.0000	0.0390	1.3520	0.0760	4.2982	0.0610	0.15800
MLR-PPC-MFL	0.1076	3.7384	0.0340	1.5247	0.0430	5.6500	0.0590	0.16200
BR	NG	NG	0.256	2.400	0.227	6.350	0.262	1.232
LP	NG	NG	0.3100	2.235	0.2670	8.065	0.2460	0.733
RAKEL	NG	NG	0.2600	1.986	0.2550	9.155	0.2370	0.593
CC	NG	NG	0.2830	1.756	0.2560	7.249	0.2680	0.619
PS	NG	NG	0.4270	2.331	0.3210	8.313	0.2870	0.845
ECC	NG	NG	0.8020	3.817	0.6850	10.731	0.7750	2.662
EPS	NG	NG	0.3000	2.138	0.2650	8.303	0.2250	0.629
ML-KNN	NG	NG	0.263	2.320	0.228	6.300	0.219	0.456
BP-MLL	NG	NG	0.318	3.150	0.235	8.005	0.821	0.7447

From Tables 11and 12, the following significant points could be noted:

- In general, and using any of the proposed or existing PTMs, MLR-PPC overcomes the existing algorithms using the four datasets, and based on the most commonly used evaluation metrics.
- MLR-PPC shows a superior performance over most other existing multi-label algorithms, when using the HAPCF or HSDF as a transformation methods.
- In general, MLR-PPC when applied using the proposed PTMs overcomes the case when applied using the existing PTMs, on the four datasets using the five evaluation metrics.
- The transformation step of a multi-label dataset into single label dataset plays an important role in the predictive performance of any multi-label classifier. Hence, this step should be considered wisely. Also, the transformation step should facilitate capturing the most accurate correlations among labels, in order to enhance the final accuracy of any multi-label classifier.

Table 13 shows the running time (in seconds) for the proposed algorithm with respect to the PTMs being used.

Table 13. Running time for the proposed MLR-PPC algorithm with respect to PTM being used.

Dataset	**Algorithm**	**Running Time (S)**
Scene	MLR-PPC-HAPCF	5.1
	MLR-PPC-HSDF	5.1
	MLR-PPC-HAPCSDF	5.17
	MLR-PPC-LFL	5.4
	MLR-PPC-MFL	6.33
Emotions	MLR-PPC-HAPCF	1.06
	MLR-PPC-HSDF	1.06
	MLR-PPC-HAPCSDF	1.06
	MLR-PPC-LFL	1.07
	MLR-PPC-MFL	1.06
Flags	MLR-PPC-HAPCF	1.0
	MLR-PPC-HSDF	1.0
	MLR-PPC-HAPCSDF	1.0
	MLR-PPC-LFL	1.01
	MLR-PPC-MFL	1.0
Yeast	MLR-PPC-HAPCF	3.52
	MLR-PPC-HSDF	3.41
	MLR-PPC-HAPCSDF	3.35
	MLR-PPC-LFL	3.49
	MLR-PPC-MFL	3.72

Figure 1 depicts the running time for the proposed MLR-PPC algorithm with respect to the three proposed PTMs (HAPCF, HSDF, and HAPCSDF) and the two existing PTMs (LFL and MFL).

From Table 13 and Figure 1, the conclusion that could be made is that the proposed PTMs showed better results in the four considered datasets in term of running time. Moreover, the HSDF showed the best running time among the proposed and the existing PTMs. The HAPCF and the HAPCSDF nearly showed the same performance on the four datasets.

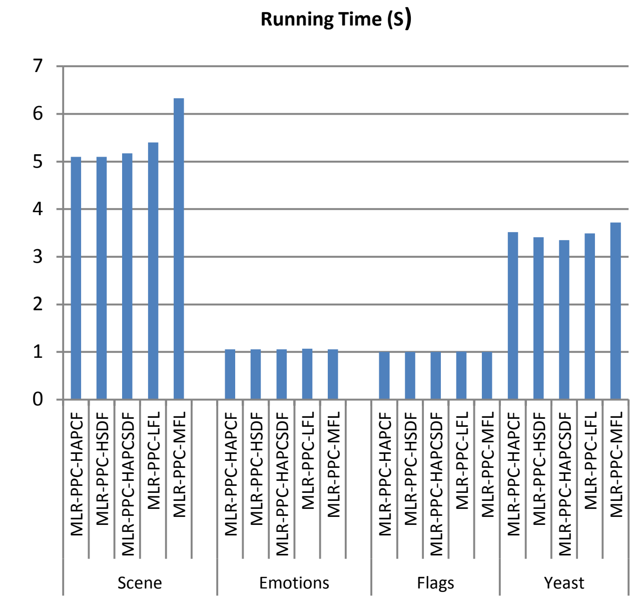

Figure 1. The MLR-PPC running time with respect to the PTM being used

5. Conclusions and Future Work

In this paper, three new PTMs have been proposed. These PTMs are based on pairwise positive correlations among labels, and not based on labels frequency as in conventional PTMs. Also, this paper proposed a second order MLR algorithm. The proposed algorithm showed a superior performance when compared to a wide variety of multi-label classification and ranking algorithms. Also, this paper showed that the proposed PTMs are better than the existing PTMs; due to two main reasons. First, the proposed PTMs do not suffer from the common problems in the traditional PTMs that depend on the frequency of labels as a transformation criterion (imbalance class distribution with the MFL and the small number of instances associated with each label when using the LFL). Second, the proposed PTMs guarantees the exploring of the most accurate positive correlations among labels, and hence, improving the predictive performance of the classification and ranking tasks.

As a future work, we intend to maximize the exploitation of the correlations among labels by proposing new high order MLR algorithms. Also, we intend to propose new PTMs that optimize the accuracy of the base classifier on the transformed dataset, as well as the accuracy of predicting the labels that have been discarded due to the transformation step.

References

[1] Alazaidah, R., Ahmad, F. K., & Mohsen, M. F. M., "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*, 12(4), pp. 218-223, 2017.

[2] Alazaidah, R., Ahmad, F. K., Mohsen, M. F. M., & Junoh, A. K., "Evaluating Conditional and Unconditional Correlations Capturing Strategies in Multi Label Classification," *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(2-4), pp. 47-51, 2018.

[3] Alazaidah, R., Thabtah, F., & Al-Radaideh, Q.,"A multi-label classification approach based on correlations among labels," *International Journal of Advanced Computer Science and Applications*, 6(2), pp.. 52-59, 2015.

[4] Al-luwaici, M., Junoh, A. K., Ahmad, F. K., Mohsen, M. F. M., & Alazaidah, R., "Open research directions for multi label learning," *In 2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*. IEEE, 2018

[5] Alzubi, O., Alzubi, J., Tedmori, S., Rashaideh, H., & Almomani, O., "Consensus-Based Combining Method for Classifier Ensembles," *International Arab Journal of Information Technology (IAJIT)*, 15(1), 2018.

[6] Fürnkranz, J., Hüllermeier, E., Mencía, E. L., & Brinker, K., "Multilabel classification via calibrated label ranking," *Machine learning*, *73*(2), pp. 133-153, 2008.

[7] Gibaja, E., & Ventura, S., "A tutorial on multi label learning," *ACM Computing Surveys (CSUR)*, *47*(3), 52, 2015.

[8] Goncalves, E. C., Plastino, A., & Freitas, A. A., "A genetic algorithm for optimizing the label ordering in multi-label classifier chains," In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pp. 469-476, 2013.

[9] Gonçalves, T., & Quaresma, P., "Using ir techniques to improve automated text classification," *In International Conference on Application of Natural Language to Information Systems*, pp. 374-379, Springer, Berlin, 2004.

[10] Quinlan, J., "C4.5: programs for machine learning," *Machine Learning*, *240*, 302, 1993.

[11] Read, J., "A pruned problem transformation method for multi-label classification," In *Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, vol. 143150, 2008.

[12] Read, J., Pfahringer, B., Holmes, G., & Frank, E., "Classifier chains for multi-label classification," *Machine learning*, *85*(3), pp. 333-359, 2011.

[13] Tsoumakas, G., & Vlahavas, I., "Random k-labelsets: An ensemble method for multilabel classification," *Machine learning: ECML 2007*, pp. 406-417, 2007.

[14] Tsoumakas, G., Katakis, I., & Vlahavas, I., "Mining multi-label data," In *Data mining and knowledge discovery handbook*", pp. 667-685, Springer US, 2009.

[15] Xu, S., Yang, X., Yu, H., Yu, D. J., Yang, J., & Tsang, E. C., "Multi-label learning with label-specific feature reduction," *Knowledge-Based Systems*, *104*, pp. 52-61, 2016.

[16] Zhang, M. L., & Zhou, Z. H., "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, *26*(8), pp. 1819-1837, 2014.

[17] Zhang, M. L., & Zhou, Z. H., "ML-KNN: A lazy learning approach to multi-label learning," *Pattern recognition*, *40*(7), pp. 2038-2048, 2007.

Raed Alazaidah is a PhD candidate in Universiti Utara Malaysia-Malaysia. He holds a Bachelor degree of computer science from Al-Albayt university-Jordan in 1999, and a Master degree in computer science (with honors) from Philadelphia university-Jordan in 2013. As academician, he involves with two main research interests: machine learning and data mining tasks, especially classification.

Farzana Ahmad is a senior lecturer at School of Computing, Universiti Utara Malaysia, MALAYSIA. She holds a Bachelor degree of Computer Science (with Honours) from Universiti Sains Malaysia in 2003 and a Master degree in Computer Science from the same university later in 2005. She pursued her Ph.D. in Computer Science (Bioinformatics) from Universiti Teknologi Malaysia in 2012 and her doctoral work involves the development of synergy network for breast cancer progression. Her passion is to understand gene regulatory network (GRN) that underlie disease progression. With such objective, she has developed a novel synergy network that able to incorporate heterogeneous data in order to predict the disease proliferations. Currently, her new interest is in neuroscience and neuroinformatics.

Mohamad Mohsin currently is the Deputy Dean of UUM CAS Student Development & Alumni Office and a senior lecturer at School of Computing, UUM College of Arts & Sciences, UUM. As academician, he involves with Artificial Intelligent (AI) related research, a part of computer science branch that studies how machine can own intelligent as human (broad AI definition!). What he do is generally to discover the secret of machine learning principle, algorithm, intelligent system, and its capability to infer human cognition and decision-making process. In deep, his main research interests are in machine learning and data mining projects that seek hidden information from huge, complex data set and finally generate/built models to ease human decision-making process. With the emergence of big data era, the data complexity and the challenge to understand them has double that makes this work more interesting.. At the moment, most of his researches are in anomaly detection and predictive modeling main in climate change studies and outbreak. Now he is engaging with artificial immune system, several bio-inspired approaches, and text mining based research.