International Journal of Engineering & Technology, 7 (4.19) (2018) 1045-1054

SPC

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper



Multi-label Classification: a survey

Vaishali S. Tidake[1]*, Shirish S. Sane[2]

[1] *Research Scholar, Matoshri College of Engineering and Research Center, Nashik, India*
Department of Computer Engineering, NDMVPS's KBT College of Engineering, Nashik, India, Savitribai Phule Pune University
[2] *Department of Computer Engineering, K. K. Wagh Institute of Engineering Education and Research, Nashik, India*
Savitribai Phule Pune University
**Corresponding author E-mail: tidake.vaishali@kbtcoe.org*

Abstract

Wide use of internet generates huge data which needs proper organization leading to text categorization. Earlier it was found that a document describes one category. Soon it was realized that it can describe multiple categories simultaneously. This scenario reveals the use of multi-label classification, a supervised learning approach, which assigns a predefined set of labels to an object by looking at its characteristics. Earlier used in text categorization, but soon it became the choice of researchers for wide applications like marketing, multimedia annotation, bioinformatics. Two most common approaches for multi-label classification are transformation which takes the benefit of existing single label classifiers preceded by converting multi-label data to single label, or an adaptation which designs classifiers which handle multi-label data directly. Another popular approach is ensemble of multiple classifiers taking votes of all. Other approaches are also available namely algorithm independent and algorithm dependent approach. Based on results produced, suitable metric is used for example or label wise evaluation which depends on whether prediction is binary or ranking. Every approach offers benefits and issues like loss of label dependency in transformation, complexity in case of adaptation, improvement in results using ensemble which should be considered during design of underlying application.

Keywords: *classification; machine learning; multi-label; supervised*

1. Introduction

Multi-label Classification (MLC) is an act of allotting a set of predefined labels to an unseen entity by observing its characteristics. It's a supervised learning approach [20]. Classification is the most popular supervised data analysis approach and machine learning is widely used for it from many decades [36]. It has been used in various applications as listed in Table 1, like text categorization, image classification, graph classification, bioinformatics, functional genomics, emotion recognition, scene classification, semantic indexing of articles, mining social media, parallel tasks and multimedia annotation and many more [1-12]. Over the last two decades, lots of research papers, books and PhD theses have been published [1-79] and various survey papers [14-20] are also available for the same.

Table 1: Reported applications of multi-label learning

Application	Reported in
Text Categorization (TC)	[1, 2, 7, 12, 15, 44]
Image Classification	[3, 4, 15]
Graph Classification	[5]
Bioinformatics	[6, 15, 67]
Functional Genomics	[7]
Emotion Recognition	[8, 15]
Scene Classification	[9, 72]
Semantic Indexing of Biomedical Articles	[1]
Understand Students' Learning Experiences	[10]
Parallel Tasks	[11]
Multimedia Annotation	[13, 15]

The remaining contents are arranged in the following way. Section 2 shows taxonomy of MLC. Section 3 describes basic approaches and methods which follow these approaches. Section 4 shows another taxonomy of MLC according to dependency. Sections 5 and 6 list various attempts done by researchers to implement MLC like feature selection, label correlation, use of clustering, natural algorithms and many more. Sections 7-9 talk about performance metrics, datasets and tools. In section 10, conclusion is presented.

2. Taxonomy of multi-label classification

Multi-label classification is classified by various researchers in different manner. In 2007, Grigorios T. and Ioannis K. [14] have categorized present MLC techniques into transformation and adaptation as shown in Table 2. The hierarchy is shown in Fig. 1 [14-20]. As the name indicates, transformation involves conversion of data from multiple labels to single label followed by single label classification (SLC). The adaptation category involves modification of basic single-label algorithm to process multiple label data directly. In 2009, Grigorios T. et al [15] further categorized transformation into various methods depending on how many labels are handled at a time. These methods use either a single label, a pair of labels, or multiple labels at a time. These three methods are termed as first, second and high-order strategy respectively by M. L. Zhang et al [19]. Also some authors mentioned one more category namely ensemble methods. Ensemble methods are formed by combining number of existing MLC

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

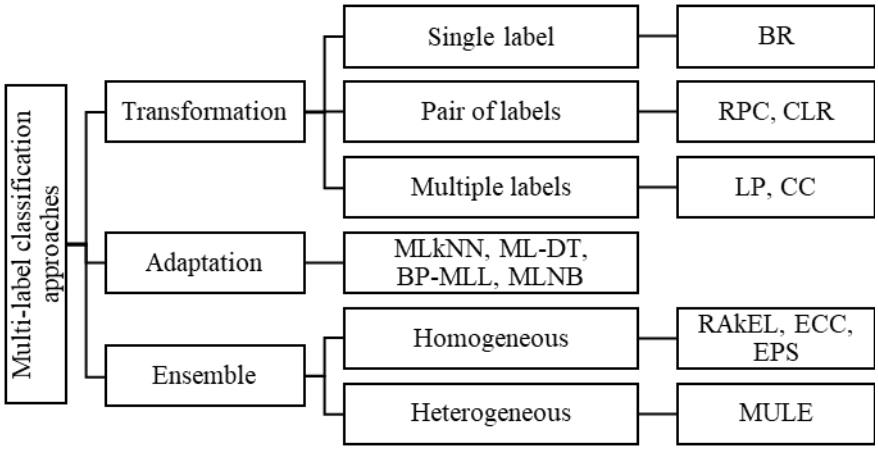


Fig. 1: Taxonomy of multi-label classification approaches

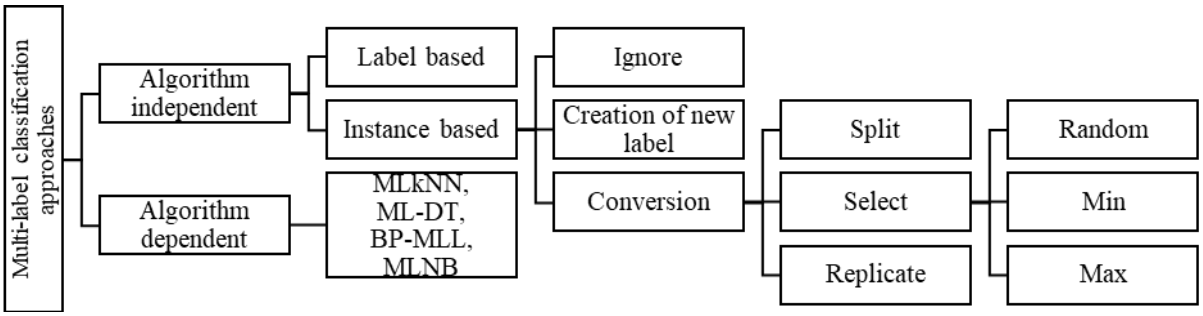


Fig. 2 Taxonomy of multi-label classification methods according to dependency

methods in different ways [15] [18]. In 2009, Andre et al [16] categorized MLC methods based on dependency of algorithm. They formed two categories namely an algorithm independent method and an algorithm dependent method as shown in Fig. 2 [14-20]. The reported literature according to taxonomy shown in Fig. 2 is listed in Table 3.

Multi-label learning can be categorized according to tasks performed during learning as shown in Fig. 3. These categories are namely classification and ranking [17] [18]. In the classification, labels are divided into relevant and irrelevant labels whereas ranking sequences all the labels in the order of relevance. One more task can be seen which combines functionality of ranking and classification [19]. According to the learning task, suitable metric can be used for evaluation discussed in section 7.

Gjorgji Madjarov et al [18] have used twelve multi-label algorithms run on eleven datasets to evaluate performance using sixteen evaluation measures. Analysis of efficiency using training and prediction time is carried out. Nemenyi and Friedman tests are used to understand statistical significance.

Table 2: Classification of reported algorithms based on approach

Multi-label classification approach	Reported in
Transformation	[6, 8, 10, 11, 14-22, 29, 30, 32-35, 37, 41-43, 49, 56, 60, 61, 63, 64, 71, 78]
Adaptation	[1, 7, 14-22, 24, 26, 29, 32-35, 37, 41-43, 56, 60, 63, 64, 71]
Ensemble	[1, 15, 18, 19, 29, 31, 35, 41, 64]

3. Basic methods, modifications, comparative discussions and weaknesses

According to taxonomy given in Fig. 1-3, the most common methods are discussed in brief in this section.

3.1 Transformation

As the name indicates, transformation involves conversion of data from multiple label to single label followed by single-label classification (SLC). It includes methods which are further classified according to the number of labels considered by classifier. These methods use either a single label, a pair of labels, or multiple labels at a time. Accordingly they are termed as first, second and high-order strategy respectively by M. L. Zhang et al [7] [9] [15] [19] [24] [28] [53] [57].

In this section some of the methods used for transformation approach are explained in brief.

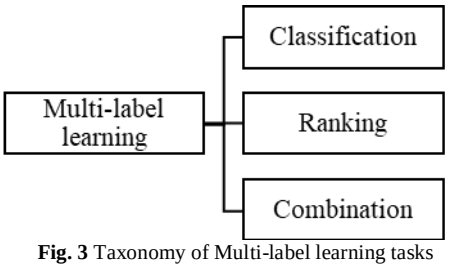


Fig. 3 Taxonomy of Multi-label learning tasks

3.1.1 Single label approaches

Methods which follow single label approach for transformation consider only one label at a time. BR and Ignore/Select are the methods which follow this approach.

Ignore/Select: These methods either remove an instance with multiple labels or select one label and associate it with that instance respectively. It is described in section 4.1 of algorithm independent methods.

Binary Relevance (BR): Consider there are three labels C_x, C_y and C_z respectively. Then BR designs three separate classifiers where each classifier handles these three labels independently. As many traditional methods are available to handle individual label,

any one method can be picked. Finally for classification of new data, results of all the three classifiers for three labels are considered. The cons of the technique is that relation among different labels is simply ignored [18-22]. But it has many good features also. As it treats each label independently, the classifier model can be easily updated dynamically if the label set is appended with a new label and scales linearly with the number of labels. Also it is beneficial to handle active data. The classifier model can run in parallel multiple classifiers of different labels. Due to so many features and ease of design, BR is very popular and widely used.

3.1.2 Pair of labels

RPC and CLR are the methods which follow transformation approach to consider two labels at a time.

Ranking by Pairwise Comparison (RPC): Let there be *m* classes in the data. Then there are *m**(*m*-1)/2 pairs of classes. For each of these pairs, a separate classifier is designed in RPC [78]. Each classifier say C_{xy} observes only those instances which have either class C_x or C_y. It ignores all the instances which have none of C_x and C_y classes or have both of them. The instance is marked as 1 or 0 if C_x or C_y is associated with it respectively. Then all the classes are ranked according to votes obtained from all such C_{xy} pair models [17] [21].

Calibrated Label Ranking (CLR): In RPC, relevant and irrelevant labels are not distinguished separately. This is achieved in CLR method which appends the label set of size *m* in the original data with a virtual (imaginary) label [78] and applies the same procedure as in RPC. As a result, all the (*m*+1) labels are ranked where relevant labels are clearly separated from irrelevant labels by an imaginary label [17-21].

3.1.3 Multiple labels

In this approach, all the labels or a group of multiple labels belonging to an instance are used. LP, RAkEL, CC and ECC use this approach. LP uses all the labels of each instance whereas CC, ECC and RAkEL use a group of labels.

Label Powerset (LP): As mentioned in section 4.1, creation of new label is one technique to deal with multiple labels an instance has. LP [17] [18] [20-22] follows similar technique. Whenever multiple labels belong to an instance, this unique combination of labels is treated as a new class. Then the data can be viewed as multiclass data which can be processed by traditional classifiers. As multiple labels are processed simultaneously, relation among labels is also considered, thereby handling drawback of BR. The problem with this method occurs when many combinations of labels appear in the original data thereby leading to many classes, where each class may be represented by comparatively less number of instances. Some classes may possess very few instances among others, which hampers accuracy. The model predicts most probable set of labels for the unseen data. Also it can predict only existing labelsets.

PPT (Pruned Problem Transformation): As seen in LP, some labelsets may possess very few instances among others, which hampers accuracy. This issue is overcome in PPT [17] [29]. Those labelsets which appear number of times less than a user defined criteria, are removed and replaced by their disjoint subsets which appear number of times less than a user defined criteria.

Random k-Labelsets (RAkEL): The complexity of LP is reduced in this method by considering only a group of labels together even if an instance has many labels associated. This group of labels is termed as a labelset. It requires to set parameter *k* which limits count of labels to be considered at a time by one model. Also it is required to set parameter *m* which denotes the number of models to be designed. It is crucial to decide parameters *k* and *m* as they affect the results. Value of *k* should be comparatively small obviously upper bounded by size of label space and *m* should not be very small. Suggested value for *m* is at least twice the number of labels [17] [19-21]. Also which combinations of labels are used by *m* models is important from results point of view. All the predictions of m models are averaged for an unseen instance. The benefit is that a non-existing label set in the original data may be predicted for the new data. G. Tsoumakas et al [21] has proposed two variations of the algorithm, one with disjoint labelsets and the other with overlapping labels respectively.

Classifier Chain (CC): As seen in BR method, three independent classifiers are designed if there are three labels C_x, C_y and C_z respectively. But this approach loses the relationship among labels if any. This problem is removed in CC where these three labels are chained in particular sequence [22]. Let the sequence be C_y, C_x and C_z. Then a classifier is designed by considering all the features to predict C_y. Then another classifier is designed by considering all the features and predicted C_y to predict C_x. Next another classifier is designed by considering all the features and previous predictions to predict C_z. Thus the process is totally serialized by a particular sequence of labels thereby considering relations among labels and hence cannot be parallelized. Finally for classification of new data, results of all the three classifiers for three labels are considered. Permutation of labels can be obtained in various ways [19] [20] [22]. New measure called log loss is also introduced in [22]. It is related to grading error by the certainty at which it is predicted. Prediction of false positives having less certainty presents logarithmically smaller penalty than prediction having high certainty. Probabilistic classifier chains (PCC) [39] is also suggested by Jesse Read et al which uses the probabilistic output of classifier yielding posterior probability computed using Naïve Bayes.

Ensemble of Classifier Chain (ECC): As seen in CC method, the sequence of labels is crucial for getting good accuracy. It is difficult to find out which sequence is the best one. Hence Jesse Read et al [22] solved this issue by combining results of number of CC models run with different sequences of labels, thereby getting better accuracy as compared to CC. One more important feature of ECC is that it surely predicts output and never empty set due to different chain sequences.

Hierarchy of multi-label classifiers (HOMER): One challenge in MLC is the scalability of algorithm with respect to dimensions of the label space. Because of more labels, the algorithm has to suffer from the class imbalance problem, computational cost of training and the inefficiency for applications requiring fast response times. To handle this problem, first root node is constructed which consists of all the labels. Next clustering with balanced k-means [23] is employed to divide labels into clusters which represent new nodes. Classifier for each cluster is designed to handle labels in that cluster only. If the predicted label is in meta-labels of the child node, then only classifier of that child node is called. Advantage of balanced clustering is that the related labels belong to the same cluster, hence same node of the tree. So only classifier of that node need to be invoked thereby reducing cost of prediction. Also each node handles less training instances thereby improving predictive performance. Note that clustering of labels is done by G. Tsoumakas et al by partitioning labels into clusters and tree structure is used for representation [17] [20].

3.2 Adaptation

The adaptation category involves modification of basic single-label algorithm to process multiple label data directly. MLC is based on supervised learning approach and many machine learning algorithms are already available for supervised learning. Many researchers have used these machine learning algorithms with necessary modification to suit for multi-label data in the past and still there is a need for the research in this area.

These methods consists of algorithms suitable to deal with the multiple label data. M. L. Zhang et al [19] described these methods as they “fit an algorithm to the data”. These methods have tuned the basic classifiers like decision tree, support vector machine, Naïve Bayes, neural network and k nearest neighbors to suit multiple label data without conversion [14-20].

In this section some of the methods used for algorithm adaptation approach are explained in brief.

Multi-Label k Nearest Neighbours (ML-kNN): It is an adaptation of conventional kNN to handle multiple label data [24]. For an example in the dataset, *k* neighbors are computed using Euclidean distance. For each label C_m of an example X_n, number of neighbors having label C_m is counted. For each label C_m not belonging to an example X_n, but still having neighbors with label C_m are counted. Using these counts, likelihood probability is computed. From the training set prior probabilities are also obtained by counting examples having label C_m and not having label C_m respectively. Next labels of new example are obtained using MAP (Maximum a posteriori) whose base is Bayes theorem [12] [10]. Using prior and likelihood probabilities, posterior probability to have label C_m is computed for an unseen example, given exact count of neighbors having label C_m. MLkNN [17-21] [24] has shown better results compared to various algorithms though it has one drawback of not considering relationship between labels.

AdaBoost.MH: Boosting [48] [70] is the process which assigns weights to training examples. Training <example, label> pairs which are tough to classify correctly are assigned higher weights whereas those which are easy to predict are assigned lower weights. It forces the learner to give attention to those labels and examples which will prove to be the most beneficial to get a highly accurate classification rule. R. Schapire and Y. Singer [25] proposed AdaBoost.MH algorithm to handle multiple labels. As output of all classifiers cannot be given equal importance, their weights are obtained to increase the performance of classifiers. AdaBoost.MH aims to minimize the hamming loss. It predicts all of the correct labels using classification [17] [18] [21] [25].

Backpropagation Multi-label neural network (BP-MLL): It's a modification of conventional neural network with feed-forward for handling multiple label data. M. L. Zhang et al [7] have designed global error function by considering label correlation. In this function, m^{th} error term gives network error on (x_m, y_m) instance. It computes how much output of the network differs on relevant and irrelevant labels of x_m. Bigger the difference, the performance is better. Negative value of this difference is given to exponential function. This error function which is global, is minimized by combining backpropagation and gradient descent. As a future work, authors have mentioned ensemble of BP-MLL to improve performance [17] [18] [21].

ML-C4.5: A. Clare and R. King [26] modified traditional C4.5 decision tree [27] algorithm for handling multiple labels. They also introduced technique for feature selection in multi-label data namely ML-IG. It finds entropy for each class C_x using the information of probability of class C_x. It uses the information of instances which belong to class C_x and instances which do not belong to class C_x respectively. Such probabilities are calculated for each class. This information is used for feature selection followed by C4.5. The difference is that leaves of the tree are assigned set of labels and not a single label. As a result, stable and accurate rules are generated for many labels at one and two levels in the tree. At three and four levels, no useful rules were found. The algorithm follows transformation approach [16] [18] [19] [21] [26] [27].

MLNB: Zhang M. L. et al [28] have done adaptation of Naïve Bayes to suit multi-label data. First for each class C_x, the prior and conditional probabilities are computed which are then used to estimate the posterior probabilities. The problem with this algorithm is that the correlation between labels is not considered and this may affect its performance.

3.3 Ensemble

It is observed that sometimes single classifier cannot predict with the expected accuracy level. But accuracy level improves if the same classifier is run several times with significant variation of some parameter and then the obtained results are combined. This approach is used by RAkEL [18] [21] and ECC [18] [22]. As described in section 3.1.3, RAkEL is an ensemble of several LP classifiers. Ensembles of classifier chains (ECC) is a technique that uses several classifier chains as base. Ensemble approach can further be classified as homogeneous or heterogeneous [1]. RAkEL, ECC and EPS (Ensemble of Pruned Sets) [18] [30] are homogeneous ensembles as they use same type of base classifiers namely LP, CC and PS respectively. MULE [1] is heterogeneous ensemble as it uses different types of base classifiers.

4. Multi-label classification according to dependency

4.1 Algorithm independent methods

These methods consists of those methods which can use wide range of already exiting single-label algorithms. It is possible because these methods modify their input data characteristics from multi-label to single-label [14-17] [21]. Thus the data is changed, and not an algorithm. Zhang M. L. and Zhou Z.H. [19] described these methods in very appropriate words as they "fit the data to an algorithm". These methods are further categorized to label and instance-based respectively. Let the original multi-label data consists of C labels.

A. Label-based methods: In label-based methods, C single label classifiers are designed. Each C_k classifier treats instances having label *k* as relevant and instances not having label *k* as irrelevant.

B. Instance-based methods: In instance-based methods, again variations are available according to the way used for assigning label(s) to an instance.

a. Ignore: The easiest way which simply ignores all those instances which are associated with more than one label.

b. Creation of new label: It creates new label to represent each distinct set of two or more labels appearing in the data.

c. Conversion: It converts data from multi-label to single-label. One method for conversion is to *split* the instances. For example, if there are only two labels C_m and C_n. Then data is split into D_1 and D_2. If an instance *k* has both classes C_m and C_n, then (instance *k*, class C_m) will belong to D_1 and (instance *k*, class C_n) will belong to D_2. If an instance has only one class as (instance *p*, class C_m), then it will belong to D_1 only.

Other approach for conversion is based on *selection*. When an instance *k* belongs to classes C_m and C_n, then

- Random approach selects C_m or C_n randomly and assigns to instance *k*.
- Min approach selects C_m if its occurrence is minimum between C_m and C_n and assigns to instance *k*.
- Max approach selects Cm if its occurrence is maximum between C_m and C_n and assigns to instance *k*.
- Replicate approach makes two replica of the instance as (instance *k*, class C_m) and (instance *k*, class C_n).

The problem with ignore approach is that by removing an instance with more than one label, lots of data is lost. In the alternative approach, data loss is comparatively less in case of min, max and random approaches. There is no data loss in replicate and split approaches, but the count of instances increases.

4.2 Algorithm dependent methods

These methods consists of those methods that follow the approach of designing an algorithm which suits multi-label data. These methods have tuned the basic classifiers like decision tree, support vector machine, Naïve Bayes, neural network and *k* nearest neighbors to suit multiple label data without conversion [14-21] [26-28]. Few of them are described in section 3.2. Some commonly used methods for MLC are summarized in Table 4. It assumes that the size of the label space is *m.*

5. Other MLC methods

Many attempts are done in MLC using association classification. J. Arunadevi et al [32] used associative classification along with evolutionary algorithm for MLC. Authors first perform problem transformation from multi-label to single-label. Apply Apriori algorithm for generation of rules. Use hybrid evolutionary algorithm for improvement. Ravi Patel et al [33] first convert all the nominal attributes to numeric. For example, let Salary is a field whose values are Low, Medium or High. Then each cell in the dataset with (Salary = Low) is replaced by 1, (Salary = Medium) is replaced by 2 and (Salary = High) is replaced by 3. For the other attributes, use numbers other than 1, 2 or 3. Thus each nominal attribute is replaced by a number. Next use FP-growth algorithm to generate association rules. As a future [33], it is possible to pass generated rules in genetic algorithm as an initial population and get better rules. Also heuristic search methods can be used to discover informative rules. Raed Alazaidah et al [34] transform multi-label dataset to single-label. For each instance which belongs to multiple labels, they remove all the labels except the one which is the least frequent label in that column. Discover positive correlations among labels and create rules for all the instances. For example, if labels C_x and C_y are correlated, then create rule as "if C_x = 1 then C_y = 1". Apply the rule-based classification algorithm PART on the rules created in the previous step. Last step is the prediction phase. In [35], k-means is used along with association classification. First k-means is applied for clustering of features. Count of clusters is selected from count of labels. Then each cluster S_x is represented by the label C_y which has the maximum proportion of count of instances having the label C_y in the cluster S_x to total count of instances in the cluster S_x. Next for each cluster data rules are generated. For a test instance, rule is obtained from each cluster.

E. Spyromitros et al [37] has proposed an algorithm BRkNN using lazy approach. Initially kNN is applied on the multi-label data to obtain k neighbors. Once neighbors are obtained, then BR classifier uses these neighbors independently for prediction of each label. Two variations of BRkNN are also implemented by the authors. In case there is no relevant label predicted, then first variation returns the most probable label and the second variation returns *p* most probable labels where *p* is equal to average of count of labels belonging to *k* neighbors. Benefit of these methods is that they never output empty sets. Authors have compared their methods with LPkNN and MLkNN [24].

C. Vens et al [38] have proposed three classification approaches based on the decision tree: HMC which considers all the super classes of a node using mean, SC which constructs separate tree for each class and HSC which considers conditional probability of class C with its parent. HMC and HSC can also be used for classification using DAG (Directed Acyclic Graph) whereas SC is not applicable to DAG. Authors have used AUPRC (Area Under Precision Recall Curve) for evaluating prediction performance. Main contribution of authors is the use of class hierarchy DAG which is not studied earlier [38].

Classifier chains introduced by Jesse Read et al [22] uses the greedy algorithmic strategy. It only searches for the most probable label combination. But if all the label combinations are searched for, then definitely the best result is obtained. This approach is used by PCC [39] which computes the conditional probability for every label set based on the product rule of probability. Disadvantage is that the complexity is high at the time of prediction. Authors [39] have used risk minimization model to minimize rank loss, subset 0/1 loss and hamming loss. Ensemble methods ECC [22] and EPCC [39] are also used for experimentation. It is observed that the probabilistic versions PCC and EPCC are well-suited for all the three measures listed here. Also EPCC performs the best getting benefit of ensembles.

Some researchers have also used hypergraph for MLC. Spectral learning feature of Hypergraph [40] is useful to explore the correlation of labels. It's very useful for high order relations. Hypergraph, a generalization of simple graph, consists of hyper edges. Hung-Yi Lo et al [41] also uses hypergraph to capture the relation between multiple labels and the instances jointly.

[42-44] all use the same base idea. They compute the membership degree termed as the degree of relevance. Three terms namely the membership degree of each term t_x in each category C_y, that of each term t_x in each document d_z and that of each document d_z in each category C_y are computed and combined to get final membership degree. All [42-44] perform clustering to reduce computational cost of kNN and also help to reduce features.

6. Other related issues

6.1 Feature selection or dimensionality reduction in MLC

Many applications in real life use data with complex structures. For example, XML web document, chemical compounds, program flow, etc. Such data cannot be represented with feature vectors properly. In that case, graph proves to be better solution [5]. When vectors are used to represent features, then feature selection process is somewhat easier because it is assumed that all the features are available initially. This is not possible for graphs because as size of graph increases, complexity increases too much. Authors have mentioned use of label correlations for graph classification with feature selection as future scope.

Trohidis, K. and Tsoumakas, G. et al follow transformation approach in [8]. General approach for feature selection by many researchers is as follows: Convert data from multi-label to single-label. Then apply traditional single-label feature selection technique like chi-square and use max or average technique to select best features. In max technique, N number of features are selected which have maximum chi-square values. In average technique, average of all the values for each feature is obtained within all the labels weighted by prior probability of every label and then N number of features are selected having maximum values. BR can be applied on these selected features only. The problem with this method is that it considers each label independently. This issue is handled by authors [8] by using LP instead of BR. The benefit is that LP implicitly considers label correlations thereby giving better results when used with chi-square for feature selection. Authors have extracted features of two categories namely rhythmic and timbre from music using the Marsyas tool [8] followed by emotion labeling and annotation by music experts.

A. Clare and R. D. King [26] has introduced feature selection technique ML-IG to handle multiple label data as given in section 3.2. Gao, Sheng et al [45] have used Singular Value Decomposition (SVD) based Latent Semantic Indexing (LSI) for feature selection. Initially term document matrix M is decomposed into multiplication of three matrices as $M=USV^T$ where U, S and V are left singular matrix, diagonal matrix of singular values and right singular matrix respectively. Also U and V are column orthonormal. U, S and V matrices are much smaller than M. The advantage is that it greatly reduces computation requirements.

There are two ways for dimensionality reduction namely unsupervised and supervised. For example, first can be achieved using Principle Component Analysis and later can be achieved using Linear Discriminant Analysis. Y. Zhang et al [46] used basic idea which tries to identify a feature space of small size to maximize dependency between labels and features. It uses Hilbert-Schmidt Independence Criterion (HSIC) for measurement of dependency. Initially algorithm constructs label kernel matrix *L* from label space *Y*. Next eigenvectors conforming to largest *m* eigenvalues to get projection *P* from original features to the reduced features. Authors suggested a variation to use HSIC with gradient descent.

Ji S. et al [47] used least squares loss for the classification to compute the shared structure and solved a generalized eigenvalue problem. M. L. Zhang [28] has implemented feature selection with

multi-label Naïve Bayes (MLNB) algorithm. First use multi-label dataset D_o to apply PCA for feature extraction followed by genetic algorithm for feature selection. If *f, C* and *h(.)* denote feature, label and classifier respectively, then $h_f(C) = 1$ if f is selected otherwise $h_f(C) = 0$ if not selected. Form new dataset D_n from selected features. Divide D_n into ten parts and use tenfold cross validation for evaluation. Author has used the fitness function based upon the average of hamming loss and ranking loss generated by portion of dataset D_n used in all the ten folds. Next step is to apply MLNB which makes use of prior and posterior probabilities.

PPT along with mutual information is also preferred by some researchers [29] [49]. First PPT (Pruned Problem Transformation) [29] is used for data conversion and then mutual information [49] is used for feature selection. It follows transformation approach.

Li S. et al [50] used information gain for an ensemble of multi-label feature selection. Initially the dataset is partitioned into clusters using k-means. Label cardinality introduced in [14] is used to set count of clusters. Then information gain of every feature x_k with respect to each label C_k is computed and normalized. The normalized value 0 and 1 indicate that particular feature and label are independent or dependent respectively. Next using normalized values of each feature for all labels, IGS value is calculated and procedure is repeated for all the features using all instances in each cluster separately. Aggregate IGS value of each feature is computed as the summation of aggregate IGS value of that feature among all the clusters. Summation of aggregate values of all the features S is used to decide stopping criterion. All features are sorted in descending order of aggregate IGS values. These features are selected one by one till addition of their aggregate IGS value is less than threshold set and only these features are considered. $S^*\partial$ is used to set threshold. ∂ belongs to [0, 1]. Authors repeated experiments with ∂ changed from initial value 0.05, step 0.05 and final value 0.95 and found that ∂ equal to 0.35 and 0.9 give good results in text and biological domain respectively.

Li L. et al [51] used the information gain to measure degree of association between feature f_x and label C_y. Larger value represents better association. It calculates information gain IGS of each feature with respect to the whole labelset. These values are normalized and their average is used to decide threshold parameter μ. Every feature with IGS value less than μ, is removed from the list.

Jungjit and Freitas [52] used Pearson's correlation coefficient and genetic algorithm for implementation. They represented each instance by n bits string. Bit $f_x = 1$ or 0 denotes whether feature f_x is selected or not respectively. Fitness function is based upon Pearson's linear correlation coefficient. Individuals at each generation are chosen by combining tournament selection operator with elitism generator. Next crossover and mutation are carried out. Feature selection by HC (Hill Climbing) [52] is used for comparison of the results. It should be noted that genetic algorithm selects more features as number of input features increases. HC is better in this case.

Zhang, M.L. and Wu, L. [53] have not induced classifier from the original features. Instead original features are used to construct label specific features using k-means clustering, and then used for inducing classification model. That is, *m* features are represented using 2*k* clusters, *k* positive and *k* negative. Thus m-dimensional feature space is reduced to 2*k* dimensional feature space where *m>>k* in the LIFT algorithm proposed by authors. They designed two variants of the algorithm, one using information gain of all the features and the other using relation between labels and instances.

Relief [54] is one of the feature selection method used for single label learning. It rewards if two attributes have different feature value for two classes and apply penalty if two attributes have different feature value for same class. Relief for multi label [55] searches for k neighbors and also uses dissimilarity of instances to find importance of features.

Newton Spolaˆor [56] stated about BR that it converts data from multi-label to the single-label data. Then for every label, contribution of each feature is quantified and average of score of all the features within all the labels is measured. At the end features having average score above threshold are chosen. Lazy approaches are proved beneficial while evaluating methods of feature selection methods. The reason is that classifiers based on lazy approaches are generally vulnerable to irrelevant features. Three approaches of feature selection are practiced by most of the researchers namely filter which is not dependent of the learning algorithm, wrapper which is used along with the learning algorithm and the embedded approach in which feature selection is the part of training process. Different feature importance measures are used widely in the literature like information gain [79], chi square, relief, gini index, fisher score, rough set, etc. Information gain and chi square do not consider feature interaction and are the most widely used. If there are three labels C_x, C_y and C_z, then data with all the features and an individual label is constructed. Then for each feature x_k, its information gain with respect to each label is computed separately. Only if the average of all three values is greater than the threshold, then feature x_k is considered by the learning algorithm. The process is repeated for all the features. Author has used the threshold value 0.01. Also spider graph is used for experimentation and comparison is done using the R framework.

6.2 Label correlation/dependency based MLC algorithms

Label cardinality and label density introduced by Tsoumakas, G. et al [14] imply that datasets having equal label cardinality and unequal label density can possess varying characteristics and behave different for MLC methods. Former denotes average count of labels per example whereas the later denotes ratio of label cardinality to the size of label space.

[8] [32] and [40] briefed in sections 5 and 6 use label correlation.

M. L. Zhang et al [57] have used Bayesian network structure that very nicely encodes conditional dependencies of labels and feature set. It considers feature set X as the common parent of all the labels. Bayesian network constructs DAG to characterize joint probability of all labels on feature set. Then a binary classifier is designed for every label considering parent labels in DAG as added features.

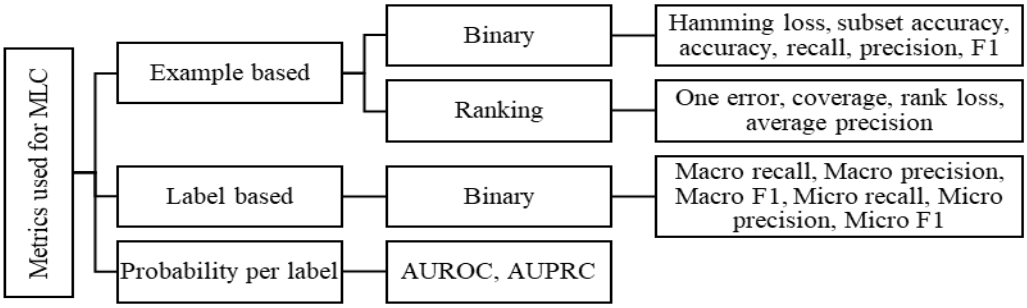


Fig. 4 Taxonomy of performance metrics [15] [18] [19]

Z. H. Zhou et al [58] used the basic idea of the relation between labels. If two labels are associated, then hypothesis generated for a label may help for the other label. Main finding of this paper is that the label relationship is asymmetric. If $R_s(m, n)$ is the reuse score from label n to m, then $R_s(m, n)$ is not necessarily same as $R_s(n,m)$. Authors implemented the boosting approach with hypothesis reuse. It produces an estimate of the label relationship as output. There are three kinds of relationships among labels possible. They are reuse score, co-occurrence relationship and Ø-coefficient relationship respectively.Basic idea behind [59] is that a label correlation may be shared by only a subset of instances rather than all the instances. Exploiting such correlations globally may be misleading and may hurt the classifier performance by predicting some irrelevant labels. The approach used is to separate training data into m groups $\{G_1 \ldots G_m\}$ where instances in the same group G_x share same label correlations. These groups are created using k-means clustering by finding similarity in label vectors, instead of feature vectors. Each group G_x represents label correlations R_x. Each G_x is represented by a prototype vector P_x. For m groups, there are m prototype vectors $\{P_1 \ldots P_m\}$. Find similarity of each instance x_k with these prototype vectors P_k to get LOC code vector $L_k = \{L_{k1} \ldots L_{km}\}$ where L_{ko} is local influence of R_o on instance x_k. Then train m regression models with the original features as input and LOC codes as outputs. For unseen instance x_u, first obtain LOC code $L_u = \{L_{u1} \ldots L_{um}\}$ using m regression models. Then obtain final label vector C_u using x_u and L_u. As a future scope, authors mention use of different clustering algorithm and different loss function.

Ying Yu has proposed two techniques MLRS and MLRS-LC in 2014 [60]. In both techniques, rough set model based on equivalence relation and equivalence classes is used. Samples are said to be equivalent if their attribute values are identical to each other. It computes neighbors of each instance X_n for each label C_m. More the neighbors with label C_m, higher is the probability of X_n related to C_m. This information is computed globally for MLRS and locally for MLRS-LC respectively. Global computation involves all the instances in the dataset and local computation involves a small subset of instances thereby resulting in better results compared to global one. The author has suggested high dimensionality reduction as a future direction. Chi square is univariate and scores each feature individually. Hence used with problem transformation generally like BR and LP.

Mutual information [61] is multivariate and useful to find joint score of relevant features. Hence mutual information is suitable for multilabel classification.

Some researchers find k nearest neighbors and use their information further in the MLC algorithm. The correlation between labels can be considered in these algorithms to get better results.

6.3 Clustering

Clustering is the most popular form of unsupervised data analysis [62]. Many researchers have used clustering to reduce computational complexity of MLC [23, 35, 40-44, 50, 53, 59]. Some of them are discussed briefly in sections 3, 5 and 6. Let's see few more.

G. Tsoumakas [63] designed CBMLC algorithm in which *k* clusters are formed using training instances where value of *k* is specified by the user. Labels are not considered during clustering. Next *k* multi-label classification models are constructed for *k* clusters independently. For a test instance, its closest cluster is searched and model of that cluster is used for classification. According to G. Tsoumakas, CBMLC is the first attempt on applying clustering analysis on the dataset before feeding the data to a classifier.

In [64] authors apply clustering on the dataset. After forming clusters of labels, new train data is constructed such that only those instances which belong to label C_x are considered for training C_x. Accordingly the train data in clusters is modified. Next PS (Prunes Set) classifier is trained with modified train data.

Zhilou Yu et al [65] run k-means algorithm several times to get correlations between labels and to confirm the chain of labels in CC. Note that clustering of labels is done, not instances.

Initially G.A. Kaminka et al [66] apply dimension reduction using orthonormalized Partial Least Squares to find direction of maximum covariance between feature space and label space using SVD. Next create clusters using k-means and use Laplacian Eigen map to learn meta-labels within each cluster. Last step is to build classifier chains over meta-labels for local model learning.

6.4 Natural algorithms

Inspiring from how various things work in nature, evolutionary algorithms are evolved. Neural network in the machine learning is inspired from the working of the neuron in our brain. Ant Colony algorithm used in artificial intelligence is inspired from the life of ants. Attempts are done to improve MLC using such natural algorithms. Some of them are listed here. [28] [52] have used genetic algorithm (GA) whereas [52] has also used Hill climbing. In [33], association classification and evolutionary algorithms are used as mentioned section 6.1. In MLOCS [67], genetic algorithm is used to improve association rules. Initially the problem transformation is applied followed by the application of single label rule mining using association rules. Next genetic algorithm is applied to obtain better rules by performing bit change either on the left side of the rule or on the right side of the rule. As mentioned in section 3.1.3, J. Read et al [22] has written that the sequence of labels is very important to get desired accuracy in classifier chain. In [68], base classifier used is CC and GA is used to find the order in which labels are used in the chain of classifier. In [69], authors use Pearson's correlation coefficient to measure dependency between feature and feature as well as feature and label, and also the mutual information to find the correlation between two labels. Algorithm is implemented using Hill climbing as well as genetic algorithm. GA is applied on features for selection.

7. Assessment of MLC algorithms

Assessment of MLC algorithms can be done based on calculation [71] or output of learner [15]. Example based metrics assess performance of each example individually whereas label based metrics assess performance of each label individually [18] [19] [21] as shown in Fig. 4 and reported in references listed in Table 5. Based on the output of learner, result can be generated in three ways: learner can predict C binary values for C labels in the label space indicating whether test instance belongs to particular label or not, it can rank C labels according to their relevance or produce C probability values for C labels respectively.

Multi-label classification can be defined as follows. Let *R* be set of (x_p, y_p) denoting a training set consisting of features and labels respectively. Aim is to find a function *g(x)*. It maps each x_p to a set y_p, where $y_p \subseteq S$. Here *S* denotes complete label set.

Let AL_i and PL_i be a set of actual labels for training instance x_i and a set of predicted labels by a classifier for the same. Let *g* denotes a classifier. Let *E* and *S* denote a test set and a set of disjoint labels respectively. Some metrics are described next.

Table 5 Performance metrics used for assessment of MLC methods

Metric	Reported in
Hamming loss	[6-9, 12-20, 23-25, 28-30, 32, 35, 37, 39, 41, 43, 46, 49, 52, 53, 56, 60, 61, 64, 66-68, 71, 72, 78]
Ranking loss	[4, 5, 7-9, 11, 13, 15, 17-20, 24, 25, 28, 39, 41, 46, 52, 53, 61, 72, 78]
One error	[7-9, 12, 13, 15-20, 24, 25, 28, 41, 46, 52, 53, 72, 78]
Coverage	[7-9, 12, 13, 15-20, 24, 25, 28, 46, 52, 53, 61, 72]
Average precision	[4, 5, 7-9, 11-13, 15-20, 24, 25, 28, 41, 46, 52, 53, 60, 72, 78]
Accuracy	[6, 10, 14, 15, 17-20, 22, 29-33, 35, 37, 44, 49, 56, 60, 61, 67, 68, 71]

Subset accuracy	[6, 11, 15, 17-20, 37, 56]
Precision	[6, 10, 11, 14, 15, 17-20, 31, 32, 67, 71, 78]
Recall	[6, 10, 11, 14, 15, 17-20, 31, 32, 67, 71, 78]
F-measure	[6, 11, 15, 17-20, 29-31, 37, 41-43, 47, 56, 60, 64, 71, 78]
ROC	[25, 40, 47, 53, 71, 78]
Macro precision, Macro recall	[18, 19]
Macro F1	[1, 2, 8, 10, 18, 19, 21, 22, 37, 56, 66]
Micro precision, Micro recall	[18, 19, 78]
Micro F1	[1, 2, 8, 10, 18, 19, 21, 23, 37, 56, 63, 66]
Macro AUC, Micro AUC	[3, 8, 19]
AUPRC	[22, 38]
Hierarchical loss	[17]
Log loss	[22]
Exact match	[66]

Recall, Precision, F-Measure and accuracy [31]:

$$Rc(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\frac{|PL_i\cap AL_i|}{|AL_i|},\ Pr(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\frac{|PL_i\cap AL_i|}{|PL_i|},$$
$$F1(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\frac{2|PL_i\cap AL_i|}{|AL_i|+|PL_i|},\ Acc(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\frac{|PL_i\cap AL_i|}{|PL_i\cup AL_i|} \quad (1)$$

Ranking loss: measures how many times an irrelevant label is ranked above the relevant labels. Small value is expected.

$$RL(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\frac{1}{|AL_i||\overline{AL_i}|}|\{(y_1,y_2)|\mu(y_1,x_i)\geq\mu(y_2,x_i)\}| \quad (2)$$

Here y_1 and y_2 belong to AL_i and $\overline{AL_i}$ respectively. For an instance *r*, value of *μ(q, r)* represents the relevance of label *q* with it. Small value is desired.

Hamming loss: measures how many times an instance and its related label is not classified correctly. Small value is expected [25].

$$HL(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\frac{|V(PL_i\Theta AL_i)|}{|S|} \quad (3)$$

where Θ denotes symmetric difference. *V(.)* = 0 if for all labels AL_i and PL_i of an instance *i* are same, otherwise it's 1.

Coverage: measures how deep the list of predicted labels should be observed for including all the relevant labels to an example. Assume the most labels appear in the beginning. Less value is good.

$$CG(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\max_{y\in AL_i}\mu(y,x_i)-1 \quad (4)$$

Average precision: computes an average of labels which are relevant and ranked better than a specific relevant label. Large value is expected.

$$AP(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}\frac{1}{|AL_i|}\sum_{y\in AL_i}\frac{|\{z\in AL_i|\mu(z,x_i)\leq\mu(y,x_i)\}|}{\mu(y,x_i)} \quad (5)$$

Subset Accuracy: computes an average to check if generated label set of an instance and its actual label set is same for all the instances [14-20].

$$SA(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}V(PL_i=AL_i) \quad (6)$$

where *V(.)* = 1 if AL_i and PL_i are same for all the labels of an instance *i*, else *V(.)* = 0.

One-error: measures how many times a predicted label at the top rank is not in the list of relevant labels of an instance. Small value is desired [14-20]. Value of *V(.)* is 0 if (.) is false, otherwise 1.

$$OE(g)=\frac{1}{|\mathcal{E}|}\sum_{i=1}^{|\mathcal{E}|}V((\arg\min_{y\in S}\mu(y,x_i))\notin AL_i) \quad (7)$$

Macro averaging and Micro averaging: A binary metric V can be measured in terms of count of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN) [14-20]. For a label *c*, macro-averaged V and micro-averaged V are computed as

$$V_{macro}=\frac{1}{|S|}\sum_{c=1}^{|S|}V(TP_c,FP_c,TN_c,FN_c)$$
$$V_{micro}=V(\sum_{c=1}^{|S|}TP_c,\sum_{c=1}^{|S|}FP_c,\sum_{c=1}^{|S|}TN_c,\sum_{c=1}^{|S|}FN_c) \quad (8)$$

8. Datasets

Different datasets are available from Mulan, MEKA and LibSVM for experimentation [73-76]. Few datasets and their domains are listed in Table 6.

Some URLs to get these datasets are:

- http://meka.sourceforge.net
- http://mulan.sourceforge.net
- https://www.cs.waikato.ac.nz/ml/proper/datasets.html
- http://mlkd.csd.auth.gr/multilabel.html
- http://slashdot.org
- http://www.imdb.com/interfaces#plain

Table 6 Datasets used by MLC methods

Dataset	Domain	Reported in
BioASQ	Biology	[1]
OHSUMED	Text	[2, 15, 22, 29, 39]
ImageNet, PASCAL	Multimedia	[4]
NCI, PTC	Biology	[5]
Yeast	Biology	[6, 7, 14, 15, 17, 18, 20-22, 24, 28-30, 35, 37, 39, 40, 53, 56, 60, 61, 64, 66, 68, 71, 78]
Reuters	Text	[7, 11, 15, 17, 21, 22, 25, 30, 31, 39, 42-44, 53, 66, 78]
Scene	Images	[9, 13-15, 17, 18, 20-22, 24, 28-30, 35, 37, 39-41, 53, 56, 60, 61, 64, 66, 68, 71, 78]
EUR-Lex	Text	[11, 15, 53]
HiFind	Multimedia	[11, 15]
Web pages	Web	[13, 15, 24, 43, 46, 52]
Genbase	Biology	[14, 17, 20, 56, 66, 68]
Medical	Text	[15, 18, 20-22, 29, 30, 39, 41, 43, 52, 56, 60, 64, 66, 68]
Mediamill	Multimedia	[15, 17, 18, 20-23, 39, 53, 63]
Enron	Web	[15, 18, 20-22, 29, 30, 39, 41, 52, 53, 56, 60, 64, 66, 68]
Emotions	Multimedia	[15, 17, 18, 20, 35, 37, 39, 56, 60, 61, 68]
FunCat, GO	Biology	[15]
Delicious	Text	[17, 18, 22, 23, 66]
tmc2007	Text	[17, 18, 21, 22, 39, 53, 71]
Corel5k	Multimedia	[18, 20, 53, 56, 60, 64, 66, 68]
Bibtex	Text	[18, 21, 22, 41, 53, 56, 66, 68]
Bookmarks	Text	[18]
Slashdot	Text	[22, 39, 53]
IMDB	Text	[22, 39]
AP Titles, UseNet	Text	[25]
CAL500	Multimedia	[41, 53, 56, 60, 68]
Language log	Text	[53]
Image	Multimedia	[39, 53]
Corel16k	Multimedia	[53, 56, 66]
Flags	Multimedia	[68]
Birds	Multimedia	[66]

9. Tools available for implementation

Different tools available for experimentation of multi-label learning are given in Table 7 [73-77]. MEKA [73] [77] is an open source library. Mulan [74] is another tool for MLC. Both are based on WEKA [75]. They provide many multi-label classifiers for researchers as well as practitioner. Mulan supports libraries which can be used in Java code. LibSVM [76] provides support for SVM. All these tools support both Comma Separated Value

(CSV) and Attribute Relation File Format (ARFF) of datasets. Traditional LIBSVM supports only single-label data. It requires some modification to handle multi-label data.

Table 7 Tools available for implementation of MLC

Tools	Reported in
MEKA	[18, 20, 73, 77]
Mulan	[6, 17, 18, 20, 37, 49, 52, 55, 56, 60, 63, 66, 68, 74]
WEKA	[18, 20, 22, 29, 30, 55, 56, 60, 63, 75]
LIBSVM	[3, 10, 11, 17, 18, 41, 47, 72, 76]

10. Conclusion

Every approach and method of multi-label classification offers benefits as well issues like loss of label dependency in transformation, complexity in case of adaptation, improvement in results using ensemble alike. This should be considered while selecting a multi-label classifier approach for underlying application. Lots of work is going on in this area. Still there is a scope for improvement of MLC algorithms using natural algorithms, parallel computing, big data and others. After doing the study of some of the multi-label research, it is observed that k nearest neighbor is very popular among researchers as a choice for base classifier [24] [37] [42] [55] [60]. One of the reasons is its simplicity. Because of its lazy nature, it is susceptible to newly added label. To find neighbors various distance metrics are available, most common being the Euclidean distance. But it is observed that the neighbors selected greatly affect the result of these multi-label classifiers. Hence proper selection of neighbors is very important.

References

[1] Yannis Papanikolaou et al, Large-Scale Online Semantic Indexing of Biomedical Articles via an Ensemble of Multi-Label Classification Models, Journal of Biomedical Semantics 2017 8:43

[2] Rafal Rak et al, Multi-label Associative Classification of Medical Documents from MEDLINE, Proc. of the Fourth International Conf. on Machine Learning and Applications (ICMLA'05), 2005 IEEE

[3] Zheng-Jun Zha et al, Joint Multi-Label Multi-Instance Learning for Image Classification, 978-1-4244-2243-2/08/$25.00 ©2008 IEEE

[4] Qinghua Yu et al, Combining local and global hypotheses in deep neural network for multilabel image classification, Neurocomputing 235 (2017) 38–45

[5] Kong, X. et al, Multi-label feature selection for graph classification. In Data Mining (ICDM), 2010 IEEE 10th Int. Conf. (pp. 274-283)

[6] Ricardo Cerri et al, Comparing methods for multi-label classification of proteins using machine learning techniques, Springer 2009

[7] M. L. Zhang, Z. H. Zhou, Multi-label neural networks with applications to functional genomics and text categorization, IEEE Transactions on Knowledge and Data Engineering 18(10)(2006)1338–1351

[8] Trohidis, K. et al, 2008, September. Multi-Label Classification of Music into Emotions. In ISMIR (Vol. 8, pp. 325-330)

[9] Z. H. Zhou, M. L. Zhang, Multi-Instance Multi-Label Learning with Application to Scene Classification, Advances in Neural Information Processing Systems, 2006, pp. 1609-1616

[10] Xin Chen et al, Mining Social Media Data for Understanding Students' Learning Experiences, IEEE Transactions On Learning Technologies, Vol. 7, No. 3, July-September 2014

[11] Eneldo Loza Mencia, Multilabel Classification in Parallel Tasks, 2nd Int. Workshop on Learning from Multi-Label Data, Israel, 2010

[12] Alberto F. De Souza et al, Automated multi-label text categorization with VG-RAM weightless neural networks, Neurocomputing 72 (2009) 2209–2217

[13] Zhiqiang Zeng et al, Multimedia annotation via semi-supervised shared-subspace feature Selection, Journal of Visual Communication and Image Representation, Volume 48, October 2017, Pages 386-395

[14] G. Tsoumakas and I. Katakis, Multi-label classification: An overview, International Journal of Data Warehousing and Mining, vol. 3, no. 3, pp. 1–13, 2007

[15] Tsoumakas G., Zhang M.L., and Zhou Z.H., Tutorial on learning from multi-label data, in ECML PKDD, Bled, Slovenia, 2009 [Online].Available:http://www.ecmlpkdd2009.net/wpcontent/uploads/2009/08/learning-from-multi-label-data.pdf

[16] A. de Carvalho and A. A. Freitas, "A tutorial on multi-label classification techniques," in Studies in Computational Intelligence 205, Berlin, Germany: Springer, 2009, pp. 177–195

[17] Tsoumakas G., et al., Mining multilabel data, Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds. Berlin, Germany: Springer, 2010, pp. 667-686

[18] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, An extensive experimental comparison of methods for multi-label learning, Pattern Recognit., vol. 45, no. 9, pp. 3084–3104, 2012

[19] Zhang M.L. and Zhou Z.H., A review on multi-label learning algorithms, IEEE Transactions on Knowledge and Data Engineering, Vol. 26, No. 8, pp. 1819-1837, 2014

[20] V. S. Tidake, S. S. Sane, Evaluation of Multi-label classifiers in various domains using decision tree, Springer Nature 2018, Intelligent Computing and Information and Communication, Advances in Intelligent Systems and Computing 673, pp.117-127

[21] Tsoumakas, Grigorios, and Ioannis Vlahavas, Random k-labelsets: An ensemble method for multilabel classification, Machine learning: ECML 2007. Springer Berlin Heidelberg, 2007, 406-417

[22] Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classier chains for multi-label classification. In: Proc. of European Conf. on Machine Learning and Knowledge Discovery in Databases: Part II. ECML PKDD '09, Berlin, Heidelberg, Springer-Verlag (2009) 254-269

[23] G. Tsoumakas and I. Katakis, Effective and efficient multilabel classification in domains with large number of labels, in Proc. Work. Notes ECML PKDD Workshop MMD, Antwerp, Belgium, 2008

[24] M.L.Zhang, Z.H. Zhou, ML-KNN: A lazy learning approach to multilabel learning, Pattern Recognit., vol.40, no.7, pp.2038–2048, 2007

[25] Schapire R.E., Singer Y., Boostexter: a boosting-based system for text categorization, Machine Learning 39 (2000) 135-168

[26] A. Clare, R.D. King, Knowledge discovery in multi-label phenotype data, in: Proc. of 5th European Conf. on PKDD, 2001, pp. 42–53

[27] Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA

[28] M. L. Zhang et al, Feature selection for multi-label naive Bayes classification, Information Sciences 179 (2009) 3218–3229

[29] J. Read, A pruned problem transformation method for multi-label classification, in: Proceedings of the New Zealand Computer Science Research Student Conference, 2008, pp. 143–150

[30] J. Read et al, Multi-label classification using ensembles of pruned sets, Proc. of 8th IEEE Int. Conf. on Data Mining, 2008, pp.995–1000

[31] S. Godbole, S. Sarawagi, Discriminative methods for multi-labeled classification, in: Advances in Knowledge Discovery and Data Mining, Springer, Berlin/ Heidelberg, 2004, pp. 22–30.

[32] J. Arunadevi et al, An evolutionary multi-label classification using associative rule mining for spatial preferences, IJCA Special Issue on "Artificial Intelligence Techniques - Novel Approaches & Practical Applications" AIT, 2011

[33] Ravi Patel, Jay Vala, Kanu Patel , Classification on multi-label dataset using rule mining technique, IJRET: International Journal of Research in Engineering and Technology Vol. 03 Issue: 06, Jun-2014

[34] Raed Alazaidah et al, A multi-label classification approach based on correlations among labels, (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 6, No. 2, 2015

[35] H Haripriya et al, Multi-label prediction using association rule generation and simple k-means, 2016 International Conference on Computational Techniques in Information and Communication Technologies (ICCTICT), IEEE

[36] F. Sebastiani, Machine learning in automated text categorization, ACM Compu. Surv. 34 (1) (2002) 1-47

[37] E. Spyromitros-Xioufis, G. Tsoumakas, and I. Vlahavas, An empirical study of lazy multilabel classification algorithms, in Proc. 5th Hellenic Conf. Artif. Intell., Syros, Greece, 2008, pp. 401–406

[38] C. Vens et al, Decision trees for hierarchical multi-label classification, Mach. Learn., vol. 73, no. 2, pp. 185–214, 2008

[39] K. Dembczy´nski et al, Bayes optimal multilabel classification via probabilistic classifier chains, in Proc. 27th Int. Conf. Mach. Learn., Haifa, Israel, 2010, pp. 279–286

[40] Liang Sun et al, Hypergraph spectral learning for multi-label classification, KDD'08, August 24–27, ACM 2008

[41] Hung-Yi Lo et al, Generalized k-Labelsets Ensemble for Multi-Label and Cost-Sensitive Classification, IEEE Transactions On

Knowledge And Data Engineering, Vol. 26, No. 7, pp1679{ 1691, JULY 2014
[42] J. Jiang, S. Tsai, and S. Lee, FSKNN: multi-label text categorization based on fuzzy similarity and k nearest neighbors, Expert Syst. Appl., vol. 39, no. 3, pp. 2813-2821, 2012
[43] S. Lee et al, Multilabel text categorization based on fuzzy relevance clustering, IEEE T. Fuzzy Systems, vol.22, no.6, pp.1457-1471, 2014
[44] Rubiya P U et al, A fuzzy based approach for multilabel text categorization and similar document retrieval, IJARCSSE, Volume 5, Issue 9, September 2015 ISSN: 2277 128X
[45] Gao, Sheng et al, A MFoM learning approach to robust multiclass multi-label text categorization. In Proc. of the 21st international conf. on Machine learning, p. 42. ACM, 2004
[46] Zhang, Yin, and Zhi-Hua Zhou. Multilabel dimensionality reduction via dependence maximization. ACM Transactions on Knowledge Discovery from Data (TKDD) 4, no. 3 (2010): 14
[47] Ji, S., Tang et al, 2008, August. Extracting shared subspace for multi-label classification. In Proc. of 14th ACM SIGKDD international conf. on Knowledge discovery and data mining (pp. 381-389)
[48] J. Han, M. Kamber, Data Mining: Concepts and Techniques, The Morgan Kaufmann Series in Data Management Systems
[49] Doquire G., Verleysen M. (2011) Feature Selection for Multi-label Classification Problems. In: Advances in Computational Intelligence. IWANN 2011. Lecture Notes in Computer Science, vol 6691. Springer, Berlin, Heidelberg
[50] Li, S., Zhang, Z., Duan, J. (2014). An ensemble multi-label feature selection algorithm based on information entropy. Int. Arab J. Inf. Technol., 11(4), 379-386
[51] Li, L. et al, December. Multi-label feature selection via information gain. In International Conference on Advanced Data Mining and Applications (pp. 345-355), Springer International Publishing, 2014
[52] Jungjit et al, A new genetic algorithm for multi-label correlation-based feature selection, ESANN 2015 proc., Computational Intelligence and Machine Learning. Bruges (Belgium), 22-24 April 2015
[53] Zhang, M.L. and Wu, L., 2015. LIFT: Multi-label learning with label specific features. IEEE transactions on pattern analysis and machine intelligence, 37(1), pp.107-120
[54] K. Kira, L. Rendell, A practical approach to feature selection, Machine Learning Proceedings 1992, Pages 249–256
[55] Newton Spolaˆor et al, Relief for multi-label classification, 2013 Brazilian Conference on Intelligent Systems IEEE
[56] Newton Spolaˆor et al, A comparison of multi-label feature selection methods using the problem transformation approach, Electronic Notes in Theoretical Computer Science 292 (2013) 135–151
[57] Zhang, M.L. and Zhang, K., 2010, July. Multi-label learning by exploiting label dependency. In Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining (pp. 999-1008). ACM
[58] S.-J. Huang, Y. Yu, and Z.-H. Zhou, Multi-label hypothesis reuse, in Proc. 18th ACM SIGKDD Conf. KDD, Beijing, China, 2012, pp. 525–533
[59] Huang, S.J. and Zhou, Z.H., 2012, July. Multi-label learning by exploiting label correlations locally. In Twenty-Sixth AAAI Conference on Artificial Intelligence
[60] Ying Yu et al, Multi-label classification by exploiting label correlations, Expert Systems with Applications 41 (2014) 2989–3004
[61] Gauthier Doquire, Michel Verleysen, Mutual information based feature selection for multilabel classification, Neurocomputing 122(2013)148–155
[62] A. K. Jain, M. N. Murty, and P. J. Flynn, Data clustering: A review, ACM Comput. Surv., vol. 31, no. 3, pp. 264–323, 1999
[63] G. Tsoumakas, Clustering based multilabel classification for image annotation and retrieval, Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
[64] Pranav Gupta, Ashish Anand, Multilabel classification using label clustering, Appearing in Proceedings of the 1st Indian Workshop on Machine Learning, IIT Kanpur, India, 2013
[65] Zhilou Yu et al, An improved classifier chain algorithm for multilabel classification of big data analysis, HPCC, 2015 IEEE
[66] G.A. Kaminka et al, A scalable clustering-based local multilabel classification method, ECAI 2016
[67] Rosane M. M. et al, Multilabel OCS with genetic algorithm for rule discovery, GECCO '09 Proc. of 11th Annual conference on Genetic and evolutionary computation, pp. 1323-1330, 2009, ACM
[68] Eduardo Corrêa Gonçalves et al, Genetic algorithm for optimizing label ordering in multilabel classifier chains, 2013
[69] Jungjit S. et al, Two extensions to multilabel correlation-based feature selection: a case study in bioinformatics, Systems, Man, and Cybernetics (SMC), 2013 IEEE
[70] Quinlan, J. R. (1996). Bagging, boosting, and C4.5. In Proceedings of Thirteenth National Conf. on Artificial Intelligence, pp.725–730
[71] G. Tsoumakas et al, Random k-labelsets: An ensemble method for multilabel classification, in Proc. of 8th European Conf. on Machine Learning (ECML 2007), Warsaw, Poland, Sept.17-21, pp. 406–417
[72] M. R. Boutell, J. Luo, X. Shen, C. M. Brown, Learning multi-label scene classification, Pattern Recognition 37(9) (2004)1757–1771
[73] Read, Jesse, and Peter Reutemann. MEKA: a multi-label extension to WEKA. URL http://meka.sourceforge.net (2012)
[74] G. Tsoumakas et al, MULAN: A Java library for multi-label learning, J. Mach. Learn. Res., vol. 12, pp. 2411-2414, Jul. 2011
[75] M. Hall et al., The WEKA data mining software: An update, SIGKDD Explor., vol. 11, no. 1, pp. 10{18, 2009.
[76] C.-C. Chang and C.-J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, Article 27, 2011 [Online]. Available: http://www.csie.ntu.edu.tw/cjlin/libsvm
[77] Tidake, Vaishali S., and Shirish S. Sane. Multi-label Learning with MEKA, CSI Communications August 2016
[78] Johannes F¨urnkranz, Multilabel Classification via Calibrated Label Ranking, Mach Learn (2008) 73: 133–153
[79] S. S. Sane et al, An Effective Multilabel classification using Feature Selection, Springer Nature 2018, Intelligent Computing and Info. and Comm., Adv. in Intelligent Sys. and Computing 673, pp. 129-142