Empirical Study of Different Multi-Label Classification Methods

Apurva Dhurandher[1], Prof. Shreya Jain[1]

[1]CSE, SSTC, CSVTU, Bhilai, Chhattisgarh, India

1.Abstract – Multi-label learning is a form of learning where unlike traditional single label learning the instances can have more than one label.label.Here the classification algorithm is required to learn from a set of instances, each instance can belong to multiple classes and so after be able to predict a set of class labels for a new instance. This is a generalized version of most popular multi-class problems where each instances is restricted to have only one class label. It is gaining widespread attention due to its applicability in various areas such as semantic image analysis, text categorisation, gene functionality classification, etc. In this paper various multi-label methods are summarised and analysed. We have also presented the evaluation metrics used in the MLL setting. We have also mentioned the benchmark datasets used in the literature.

Abbreviations used –

MLL – Multi Label Learning

BR- Binary Relevance

CC- Classifier Chains

LP- Label Powerset

RAkEL - Random k-Labelsets

RPC- Ranking by pairwise comparison

CLR - Calibrated Label Ranking

ECC-ECC is ensemble of classifier chains

SVM - Support Vector Machine

PSVM- Parallel-SVM

2.Introduction –

In traditional single-label classification, the goal is to learn a classifier $h : X \rightarrow Y$ from the training set $D = \{(x_i, y_i) , 1 \leq i \leq n\}$ where $x_i \epsilon$ X (set of training instances) associated with corresponding single label $y_i \epsilon Y$ (set of disjoint labels).Based on size of label set $|Y|$, it is called binary classification (when $|Y| = 2$) or multi-class classification ($|Y| > 2$).

Even though multi-label classification was primarily motivated by the emerging need for automatic text-categorization and medical diagnosis, recent realization of the omnipresence of multi-label prediction tasks in real world problems drawn more and more research attention to this domain. For example, a text document that talks about scientific contributions in medical science can belong to both science and health category, genes may have multiple functionalities (e.g. diseases) causing them to be associated with multiple classes, an image that captures a field and fall colored trees can belong to both field and fall foliage categories, a movie can simultaneously belong to action, crime, thriller, and drama categories, an email message can be tagged as both work and research project; such examples are numerous.Hence, there is widespread rise in application of this method in different areas such as images[14,15], text[16], music[17], medical[18] and so on.

Traditional binary and multi-class problems both can be posed as specific cases of multi-label problem. However, the generality of multi-label problems makes it more difficult than the others. As opposed to single-label classification, in multi-label classification each instance is associated with set of labels Y_i where Y_i is a subset of Y. Hence, the goal is to learn a classifier h : $X \rightarrow 2^Y$ from the training set $D = \{(x_i, y_i), 1 \leq i \leq n\}$.

3. Methods –

The two popular methods in multi-label learning are - I Problem transformation method.

II. Algorithm adaptation method.

Problem transformation method transforms the learning task into single-label learning task by fitting the data to single-label algorithms, while algorithm adaptation method extends the existing algorithm for handling multi-label data directly by fitting the algorithm to given data.

I. ProblemTransformation

fit data to algorithm - transform data such that existing algorithms for binary classifier can be used.

1. The following are the simplest transformations[1][3] -

I. **Copy transformation-**The copy transformation replaces each example (x_i,y_i) with $|Y_i|$ examples (x_i, y_i), where $y_i \epsilon Y_i$. An extension to this is to use a weight of $1/|Y_i|$ to each of these newly created examples. This is called **copy-weight** method.

II. **Select** transformations- For each instance, the select family of transformation methods replaces Y_i by one of its members. There are several versions of this depending on how this one member is selected -

a. **Select-max** : Label set Y_i is replaced by most frequent label y_k in training set D, where $y_k \epsilon Y_i$.

b.**Select-min** :The least frequent label is selected for replacement.

c.**Select-random** : Here the label set is selected randomly for replacement.

III. **Ignore** transformation : - This simply ignores the multi-label instances and runs the training with single label instances only. None of these methods is likely to retain the actual data distribution and therefore is likely to have lower prediction performance.

2. **Binary Relevance (BR)**: It is one of the most popular approaches as a transformation method that actually creates k datasets (k = |Y|, total number of classes), each for one class label containing all examples of the original data set, labelled positively if the label set of the original example contained this label and negatively otherwise.
For any new instance x, BR outputs the union of the labels yj that are positively predicted by the k classifiers.
Though Binary Relevance is simple, intuitive, extremely straightforward way of handling multi-label data with low computational complexity, it ignores potential correlations among labels.Depicted in figure 1 (a).

3. **Classifier chains (CC)[3]**: This solves the above issue of label independence in BR. The basic idea of this algorithm is to transform the multi-label learning problem into a chain of binary classification problems by augmenting the input space of the next binary classifiers by the predictions of previous classifiers. This had advantage over BR methods, but the order of chaining of classifiers is still an issue here. Depicted in Figure 1 (b).

4.Label Powerset (LP)[4]- In BR and other related methods, the correlation among labels are ignored. This method preserves the correlation among labels during the transformation by considering each unique set of labels in a multi-label training data as one class in the new transformed data. Given a new instance, the single-label classifier of LP outputs the most probable class, which actually represents a set of labels.The computational complexity of LP is upper bounded by min(n, 2^m), where n is the total number of data instances and m is the total number of classes in the training data (before transformation) .Though the complexity may be high the number of actual label sets is usually much smaller in practice.The second problem with this approach is that, a large number of classes would be associated with very few examples and that would also pose extreme class imbalance problem for learning. Depicted in Figure 1 (c).

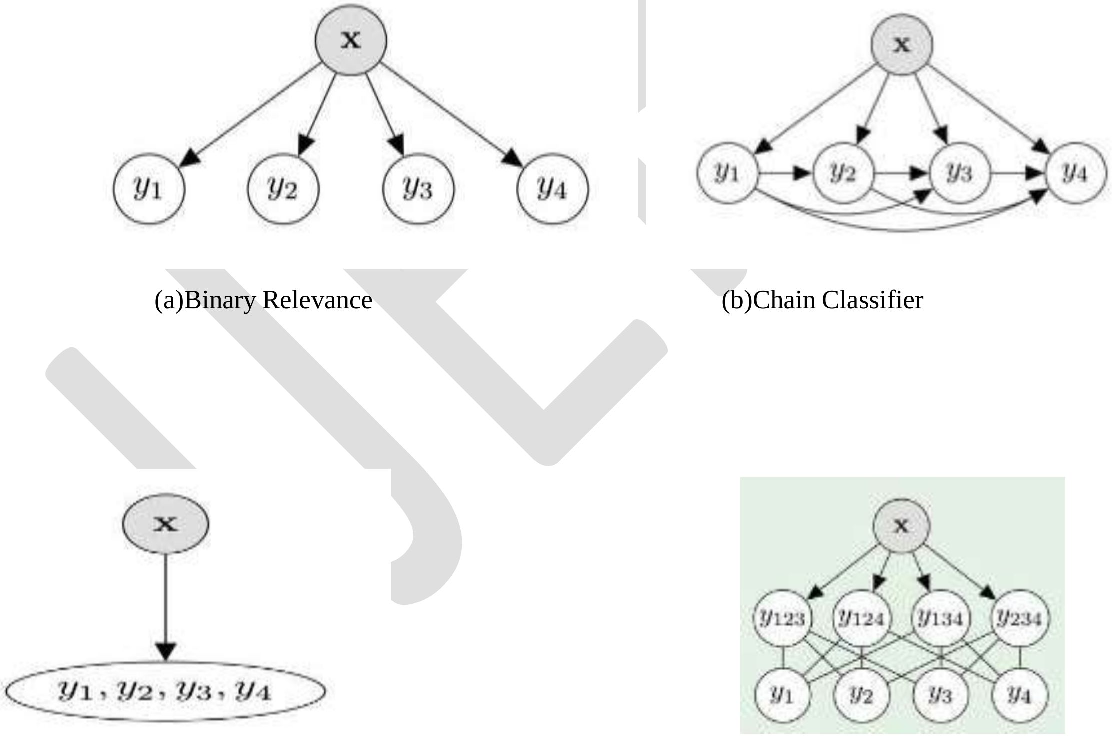
5.Random k-Labelsets(RAkEL)[5]: It solves the computational complexity and class imbalance problem of LP method .The basic idea is to randomly partition the large labelset into k smaller labelsets and for each of them train a multilabel classifier using the LP method.During prediction the output of all LP classifiers are gathered and combined. It has two variations –

(1) **RAkEL**d where the partitioned labelsets are disjoined.

(2) RAkEL$_0$ where the partitioned labelsets are overlapping.

RAkEL$_0$ has slightly better performance due to fusion of labels across different labelsets. Depicted in Figure 1 (d).

6. **Ranking by pairwise comparison (RPC[6])** : This is a ranking based method were multi-label prediction is done by first ranking the labels and then selecting the top few labels as the predicted label-set. It follows one-versus-one(OVA) approach . It transforms the multi-label data set into (m(m-1))/2 binary label data sets, one for each pair of label (y_i, y_j) where $1 \leq i \leq j \leq m$. Each dataset retains the instances from the original dataset that belong to atleast one of the corresponding labels but not both.A binary classifier is then trained on each of these datasets and ranking of labels is obtained by counting their votes for each label. Though RPC provides a relative order of the labels, but how to partition these ranked labels into relevant and irrelevant sets still remains as a challenge.



(a)Binary Relevance

(b)Chain Classifier

(c) Label Powerset

(d) Random kLabelsets(RAkEL)

Figure 1. Graphical representation of different Problem Transformation methods

II. Algorithm Adaptation –

fit algorithm to data: transform existing algorithm to use it on multi label data.

(a) **SVM-HF[7]** : It uses binary relevance method twice in two stages and uses SVMs as its base classifier. In the first stage SVMs are used as binary classifiers and in the second stage, the input space is augmented by |Y| new features. The output of first stage which are above some threshold t are marked as 1 and others as 0 in the new features assignment. Then, new set of classifiers are learnt on these new augmented training instances.For classification, similar two stage procedure is followed for the unknown test instance.

(b) **Parallel-SVM(PSVM)[8]**: It is proposed for two class multi label data and it is based on the assumption that the mixed class lies in between the two pure classes. Hence, two parallel hyperplanes are used to separate the three possible classes(class1,mixed,class2). The drawback is the number of binary classifier will increase significantly with increase in labels in training set.

(c) **Rank-SVM[9]**: It is based on ranking by support vector method. The ranking of labels is obtained by the m linear classifiers by minimizing the ranking loss. Ranking loss corresponds to average of number of label pairs from relevant set(set of the labels which belongs to the instance) and irrelevant set , which are wrongly ranked , i.e. relevant label ranked lower than irrelevant one. After ranking the output set is found using a threshold value to divide the ranked labels into relevant and irrelevant sets.

(d)**ML-*K*NN[9]** - It was one of the first instance based algorithm proposed for multi-label learning . It uses the

Bayesian inference as base classifier along with kNN to predict the labels.

$$Yt = \arg\max_{b\,\epsilon\{0,1\}} \max P\left(H_b^l \,|E^l_{ct(l)}\ , \right)$$

Here H_1^l is the event that test data **t** has label **l** and H_0 corresponds to absence of the label. Event $\underline{\mathbf{E}^l_{ct(l)}}$ denote the event that there are exactly $c_{t(l)}$ instances in N(t)(neighbourhood of t) which have label l. It uses uses the information such as membership count($c_{t(l)}$)) from kNN for prior and likelihood calculation .

(e) **Multi-label pairwise perceptron (MLPP)[11] -** It is based on a pairwise approach(RPC), i.e., it incrementally train a perceptron for each pair of classes. The classification is done based on voting method. It also suffers from quadratic complexity issue during training and testing.

(f)Backpropagation for Multilabel Learning (BP-MLL)[12][13]- This is an adaptation of the traditional multilayer feed-forward neural network under multi-label framework. The key idea is the definition of an error function, closely related to the ranking loss. The error function is minimised with gradient descent combined with the error back-propagation. The input feature, one output unit per label, and the hidden layer is fully connected with weights to the input and output layers. Its computational complexity in the training phase is high, but the time cost of predictions is quite better .

4.COMPARISON – Comparison of computational complexity . Here we compare the computational complexity of training the model of different methods mentioned above with respect to number of class labels m. We can see the label transformation methods such as **BR** and **CC** performs better in terms computational complexity but **BR** ignores the label dependencies and **CC** depends on its chaining order for better performance, hence making these two methods suboptimal. While **LP** considers all label dependencies its exponential complexity makes its unsuitable for task with high number of labels. Theoretically **RA*k*EL** has exponential complexity but in practise it is quite computationally inexpensive. **RPC, PSVM** and **ML-kNN** can be placed in between other methods in terms of computationally complexity as well as efficiency.

Methods	Complexity
BR	O(m)
CC	O(m)
LP	O(2^m)
RPC	O(m^2)
PSVM	O(m^2)
SVM-HF	O(m)
RA*k*EL	O(2^m)
ML-kNN	O(m^2)

5. Experimental Results-

5.1 Evaluation metrics

In multi-label classification, predictions for an instance is a set of labels and, therefore, the prediction can be fully correct, partially correct(with different levels of correctness) or fully incorrect. Evaluation metrics of single label classification does not capture the partial correctness of a model as their isa single output. Hence we need a method to capture such prediction with different levels of correctness. The following are the metrics used in our experimentation. $\mathbf{Y_i}$ and $\mathbf{Z_i}$ are given and predicted label sets, respectively and **n** is the number of instances.

(a)**Accuracy(A)**: Accuracy is defined as average proportion of the predicted correct labels to the total number

(predicted and actual) of labels.

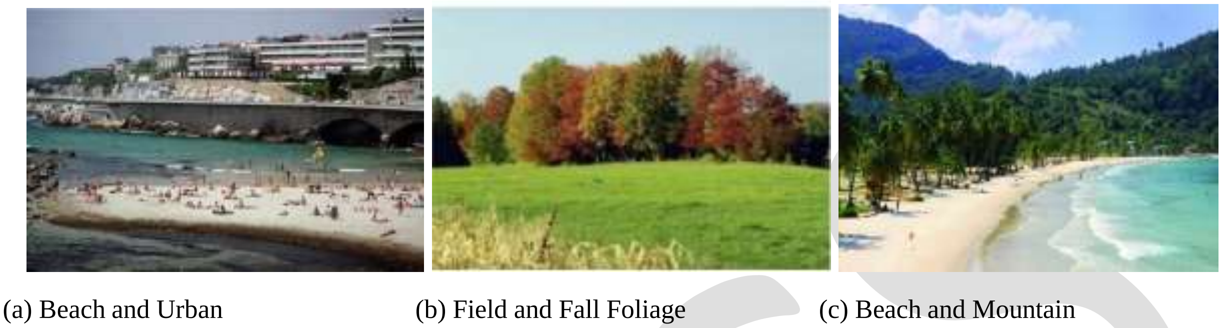
(b) **Recall (R):** Recall is the proportion of predicted correct labels to the total number of predicted labels, averaged over all instances.

(c) **F1-Measure (F):** F1-Measure is the harmonic mean of precision and recall, as followed from single-label classification.

5.2 Datasets

1.Scene Dataset Scene dataset was created to address the problem of emerging demand for semantic image categorization. Each instance in this dataset is an image that can belong to multiple classes This dataset has 2407 images each associated with some of the six available semantic classes (beach, sunset, fall foliage, field, mountain, and urban).

:



(a) Beach and Urban (b) Field and Fall Foliage (c) Beach and Mountain

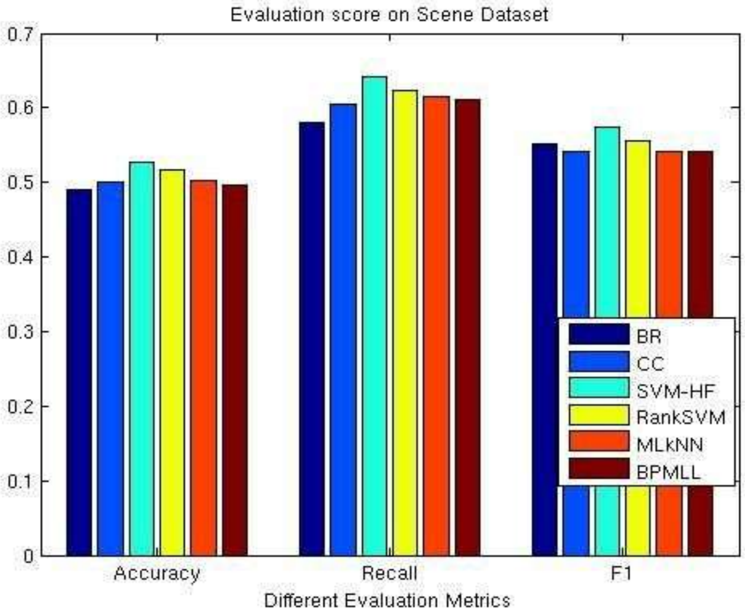
Figure 2 : Example multi-label images from Scene Dataset

2. **Yeast Dataset** In yeast data, each instance is a yeast gene described by the concatenation of micro-array expression data and phylogenetic profile. Each of these 2417 genes is associated with one or more of 14 different functional classes.

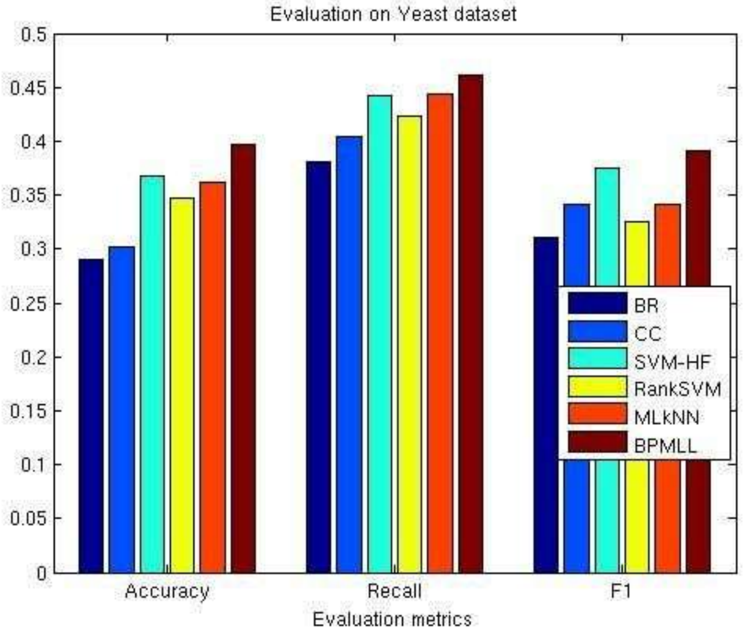
5.3 Results

We implemented some of the methods discussed above for empirical analysis. We used Matlab platform for our task. We can observe that overall accuracy of Scene dataset is more than Yeast datasets as the number of class labels in Scene is less than Yeast and hence in Yeast our classifiers makes more error than Scene.Also we can observe that different methods perform better to different datasets as evident from the results below.

We used the above two mentioned datasets(Scene and Yeast) for our experiments. We used various multi-label classification methods and compared their performance using the three evaluation metrics mentioned above. The evaluations scores are plotted against different methods for easy comparison. We used Matlab’s Bar plot tool to plot these graphs.



(a)Evaluation results on Scene Datasets



(b)Evaluation results on Yeast Datasets

7.FUTURE WORK AND CONCLUSION

In this paper, an empirical study of different multi-label algorithms, their applications and evaluation metrics has been presented. A sparse set of existing algorithms has been organized based on their working principle and a comparative performance analysis has been reported. This study provides useful insights on the relationships among different algorithms and directs light for future research. A few possible future challenges have also been identified:-

i) While it has been established that exploiting the label correlations is an important factor for improving the performance, this idea in most cases has been used intuitively; a future challenge, therefore, is to theoretically explore the conditional and unconditional dependencies and correlate the performance improvement with each kind of dependence modelling.

ii) A number of methods have been quite successful to model small and medium sparse data with reasonably good performances, however, further research attention is needed especially to deal with complicated and large data

iii) The computational complexities of most of the algorithms suggest that more efficient algorithms would be needed to achieve scale independence.

iv) Although it has been established that the data properties such as label cardinality can strongly affect the performance of a multi-label algorithm, there is no systematic study on how and why the performance varies over different data properties; any such study would be helpful to decide on multi-label algorithms for any particular domain.

v) With emerging needs for online algorithms, an important future direction would be to design efficient online multi-label approaches that scale with large and sparse domains.

REFERENCES:

[1] A Review on Multi-Label Learning Algorithms -Min-Ling Zhang, Member, IEEE and Zhi-Hua Zhou, Fellow, IEEE IEEE Transactions on Knowledge and Data Engineering , Vol. 26 , No. 8, August 2014.

[2] Random k-Labelsets for Multilabel Classification - Grigorios Tsoumakas, Member, IEEE, Ioannis Katakis, and Ioannis Vlahavas, Members of IEEE, IEEE Transaction on knowledge and data engineering , Vol. 23 , No.

7, July 2011.

[3] Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification Machine Learning 85(3), 1-27 (2011).

[4] Tsoumakas, G., Katakis, I., Vlahavas, I.: Data Mining and Knowledge Discovery Hand- book, Part 6, chap. Mining Multi-label Data, pp. 667-685. Springer (2010).

[5] Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Machine Learning 85(3), 1-27 (2011).

[6] Boutell, M., Luo, J., Shen, X., Brown