Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 115 (2017) 572–579

Procedia Computer Science

www.elsevier.com/locate/procedia

7th International Conference on Advances in Computing & Communications, ICACC-2017, 22-24 August 2017, Cochin, India

Multi-label Classification: Problem Transformation methods in Tamil Phoneme classification

M. Pushpa[a]*, S. Karpagavalli[b]*

[a]Department of Computer Science, PSGR Krishnammal college for women, Coimbatore-641004, Tamilnadu, India

Abstract

Most of the supervised learning task has been carried out using single label classification and solved as binary or multiclass classification problems. The hierarchical relationship among the classes leads to Multi- Label (ML) classification which is learning from a set of instances that are associated with a set of labels. In Tamil language, phonemes fall into different categories according to place and manner of articulation. This motivates the application of multi-label classification methods to classify Tamil phonemes. Experiments are carried out using Binary Relevance (BR) and Label Powerset (LP) and BR's improvement Classifier Chains (CC) methods with different base classifiers and the results are analysed.

Peer-review under responsibility of the scientific committee of the 7th International Conference on Advances in Computing & Communications.

Keywords: Supervised learning; Multi-label classification; Speech recognition; Phoneme classification.

1. Introduction

Classification is a commonly used data mining task. Single label (SL) classification where each instance is associated with a class has been used in several distinct domains. In certain domains such as text categorization, annotation of image, audio and video, bioinformatics, emotion recognition systems where instances may belong to one or more classes. The set of techniques that can handle instances having multiple labels has been developed and are called multi-label (ML) classification [1].

Several different methods have been developed and reported to perform multi-label learning task. Two broad

* Corresponding author. Tel.: 9787525777, 9842105105; fax: +0-000-000-0000 .
E-mail address: ranjthapushpa92@gmail.com, karpagavalli@psgrkc.ac.in

1877-0509
Peer-review under responsibility of the scientific committee of the 7th International Conference on Advances in Computing & Communications
10.1016/j.procs.2017.09.116

categories used to perform multi-label learning are the problem transformation methods and the algorithm adaptation methods. The problem transformation approach involves transformation of an input instance into a representation suitable for traditional single-label classifier. In this approach, the multi-label data representation is transformed into a single-label data representation which is acceptable by traditional SL classification methods [2]. There are various algorithms which come under this approach are BR, LP, CC, RPC, CLR, etc. The algorithm adaptation approach involves modification of an existing SL classifier algorithm and making it suitable to handle multi-label instances. Many algorithms such as MLkNN, ML-BPNN, ML-DT uses this approach. Ensemble methods are also used for multi-label learning that combines outcomes from several classifiers based on either problem transformation or algorithm adaptation. Algorithms such as RAkEL, Ensembles of classifier chains (ECC), follow this approach.

Many application areas are present for multi-label classification like text categorization, scene classification, face recognition, drug discovery, music recommendation. This work employs multi-label classification approach in Tamil language phoneme classification task.

2. Phonemes in Tamil Language

Phoneme is the most commonly used sub-word unit in speech recognition application. Design and development of efficient phoneme recognition will lead to high accuracy speech recognition applications. The phonetic units of a Tamil language can be grouped into different categories such as vowels, plosives, fricatives, etc. According to the place and manner of articulation, phonemes can be organized in hierarchical fashion [3].

Tamil phonemes are majorly classified into obstruent and sonorant categories. Obstruent is a consonant sound such as that is formed by obstructing airflow, causing a strong gradient of air pressure in the vocal tract. Sonorant is a speech sound that is produced with continuous, non-turbulent airflow in the vocal tract. The hierarchy of phonemes in Tamil language is shown in figure 1. Stop consonants also known as plosives, are consonant in which the vocal tract is blocked so that all airflow ceases. Fricatives are consonants produced by forcing air through a narrow channel made by placing two articulators close together. This turbulent air flow is called frication. Affricates are combination of stops and fricatives, i.e., begins as a stop and release as a fricative with the same place of articulation. All these stops, fricatives and affricates fall in the group of obstruents. Nasals are allowing the air to escape freely through the nose but not through the mouth, as it is blocked by the lips or tongue. In terms of acoustics, nasals are sonorants, meaning that they do not significantly restrict the escape of air. In addition to vowels, phonetic categorizations of sounds that are considered sonorant include approximants, taps and trills [4][5].

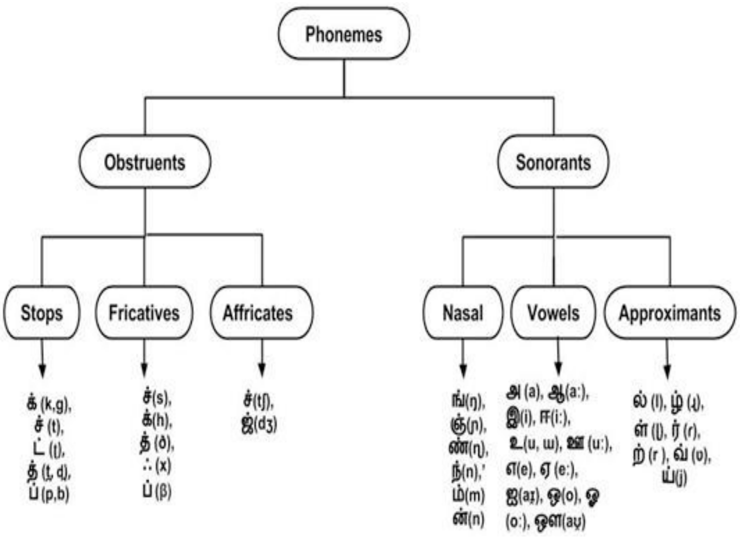

Fig. 1. The hierarchical relationship of Tamil Phonemes

In phoneme classification, in flat classification model, search space complexity and time complexity is high. The relationship among the phonemes leads to hierarchical classification which greatly reduces search space complexity compared to flat classification model [4]. In this paper, the phoneme classification task employs the multi-label classification approach to overcome the issues related to complex flat phoneme or hierarchical phoneme classification.

3. Related work

Many multi-label classification based research were carried out to solve problems in different domain. A review of few research works are presented in this section. A multi-label classification approach was attempted in text categorization and functional genomics applications using Back-propagation for Multi-Label Learning (BP-MLL). It is observed that the performance of BP-MLL is superior to those of some well-established multi-label learning algorithms [6]. In another text categorization work, a novel joint learning algorithm that allows the feedbacks to be propagated from the classifiers for latter labels to the classifier for the current label was proposed. Experiments were done with real-world textual data sets like Slashdot, medical, Enron, and tmc2007 and the results were analysed with evaluation metrics such as hamming loss, exact match, and F-score [7]. A task of emotion recognition from music was carried out as a multi-label classification task [8], because a piece of music may have more than one emotion at the same time. They proposed the Binary Relevance based Least Squares Twin Support Vector Machine (LSTSVM) multi-label classifier for emotion recognition from music. The performance of the proposed classifier is compared with the eight existing multi-label learning methods using fourteen evaluation measures.

A multi-label support vector machine active learning method was used to solve multi-label image classification problem [9]. In [10], the authors proposed Hypotheses-CNN-Pooling (HCP) for multi-label classification of images, where large single label image datasets are trained by CNN. In bioinformatics, for protein classification, some multi-label classification methods were applied [11]. They used both algorithm independent approach (RAndom k-LabEL (RAKEL) method) and algorithm dependent approach (ML-kNN) with evaluation metrics accuracy, precision, recall, F-measure and sub-accuracy. They used yeast and protein datasets and analyzed the performance of the One-Against-All, Label-Powerset, Cross-Training, ML-kNN and RAKEL.

4. Multi-Label Classification

In data mining, the classification tasks can be classified according to the number of labels to be predicted for each instance into two categories as single-label classification and multi-label classification. Single label classification refers to the classification task where there is only one label to be predicted. The basic principles of multi-label classification are similar to single-label classification, but the multi-label classification has two or more concept labels to be predicted.

The multi-label classification can be mathematically represented as, X be the domain of instances to be classified, Y be the set of labels, and H be the set of classifiers for f: $X \rightarrow Y$, where f is unknown. The goal is to find the classifier $h \in H$, maximizing the probability of $h(x) = y$, where $y \in Y$ is the ground truth label of x [12]. The attribute–value to deal with multi-label problems can be presented in another way as, A dataset is characterized by N instances $z_1, z_2, \ldots, z_N$, each containing m attributes $X_1, X_2, \ldots, X_m$ and c labels $Y_1, Y_2, \ldots, Y_c$. If i refers to the i-th instance (i=1, 2, ...,N), x_{ij} refers the value of j-th attribute (j = 1, 2, ...,m) of instance i, and output y_{ik} refers to the value of k^{th} label (k = 1, 2, ...,c) of instance i. The instances are tuples $z_i = (x_{i1}, x_{i2}, \ldots, x_{im}, y_{i1}, y_{i2}, \ldots, y_{ic}) = (x_i, y_i)$ also denoted by $z_i = (x_i, y_i)$, where the fact that z_i, x_i and y_i are vectors is implicit. Each y_i is a member of the set $Y_1 \times Y_2 \times \ldots \times Y_c$; without loosing generality we will assume $Y_i \in \{0, 1\}$, i.e., each label will only assume binary values.

In general, multi-label classification methods fall into two main categories which are problem transformation (PT) methods and algorithm adaptation (AA) methods. The problem transformation methods transform the original problem into one or more single-label classification or regression problems. The algorithm adaptation methods do not transform the problem, but rather they adapt the learning algorithms themselves to handle multi-label data. [13][14]. Since the flexibility of problem transformation methods is good, this work carried out using PT methods such as Binary Relevance (BR), Label Powerset (LP) and BR's improvement Classifier Chains (CC). Many base classifier algorithms are used in the problem transformation methods like J48, NB, SMO, AdaboostM1, ZeroR and Bagging.

4.1 Multi-Label Classification Methods

Binary Relevance (BR) Classifier

The baseline approach, called the binary relevance method extents to independently training one binary classifier for each label. In this method, a multi-label problem is converted into |L| number of binary SL classification problems where L is a set of labels. Each of the binary classifiers votes separately to get the final result. This method of dividing the task into multiple binary tasks has something in common with the one-vs.-all (OvA, or one-vs.-rest, OvR) method for multiclass classification. Note though that it is not the same method in binary relevance we train one classifier for each label, not one classifier for each possible value for the label.

Label Power set (LP)

The most natural approach is the label power set method which generates a new class for every combination of labels and then solves the problem using multiclass classification approaches. The main drawback of this approach is the exponential growth in the number of classes, leading to several generated classes having very few labeled instances leading to overfitting. BR does not consider relationship between labels. This drawback of BR is overcome by LP, also called as LC (Label Cardinality).

Classifier Chain (CC)

Classifier chains method, which is based on the BR method, overcomes the disadvantages of BR and achieves higher predictive performance, but still retains important advantages of BR, most importantly low time complexity. CC offers a general problem transformation method that inherits the efficiency of BR and competes with the high accuracy of more computationally complex methods.

The Classifier Chain approach like LP, also try to overcome the drawback of BR. Similar to BR, a ML problem is transformed into |L| number of SL problems where L denotes a set of labels and for each label Lj, a separate binary classifier Cj is designed. But the input for each classifier Cj is different. Like LP classifier, CC also needs selection of base classifier, uses J48 as base classifier by default.

4.2 Multi-Label Classification Metrics

The performance metrics of multi label classifiers can be classified as label-based and example-based. The label-based metrics are calculated for each label and then they are averaged across all labels (ignoring relations between labels) while the example-based metrics are calculated for each test example and then averaged across the test set. In the proposed work, two label-based measures, one-error, and average precision as well as two example-based measures accuracy and hamming-loss are used to evaluate the performance of the ML classifiers. Assume that D being the multi-label dataset, L the full set of labels used in dataset, Yi the subset of predicted labels for the i^{th} instance, and Z_i the true subset of labels. Accuracy is defined as the proportion of correctly predicted labels with respect to the total number of labels for each instance.

$$Accuracy = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Yi \cap Zi|}{|Yi \cup Zi|} \quad (1)$$

Hamming loss is the most common evaluation metric in multi-label literature, computed as the symmetric difference between predicted and true labels and divided by the total number of labels in the MLD. The smaller the value of hamming loss, better the performance. The performance is perfect when hamming loss = 0.

$$ham\min g - loss = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{|Yi\Delta Zi|}{|L|} \quad (2)$$

One error is a metric which determines how many times the best ranked label given by the classifier is not part of the true label set of the instance. The best performance is reached when one-error is equal to 0. The smaller the value of one-error is, the better performance.

$$OneError = \frac{1}{|D|}\sum_{i=1}^{|D|}\begin{bmatrix}\arg\max\langle rank(xi, y)\rangle \notin Yi \\ y \in Zi\end{bmatrix} \quad (3)$$

Average precision computes the proportion of labels ranked ahead of a certain relevant label. The goal is to establish how many positions have to be traversed until this label is found. The best performance is reached when average precision is equal to 1. The bigger the value of average precision is, better the performance.

$$Average\_precision = \frac{1}{|D|}\sum_{i=1}^{|D|}\frac{1}{|Yi|}\sum_{y\in Yi}\frac{|\{y'\in Yi : rank(xi,y')\leq rank(xi,y)\}|}{rank(xi,y)} \quad (4)$$

5. Experiment and Results

A speech corpus was prepared by using 15 native speakers of Tamil, where each speaker uttered each word 5 times. Nearly 100 words of Tamil language covering almost all phonemes in Tamil language have been used in recordings. In this work, Audacity digital audio editor and high quality microphone was used in recording the data. The recordings carried out in quiet room environment and at sampling rate 16 KHz with 16 bit PCM.

The phonemes are manually segmented from each speech utterance and for each phoneme, only 9 frame length spectral information is used for processing. In feature extraction, the most effective and commonly used feature extraction method, Mel-frequency Ceptral Co-efficients (MFCC) has been employed and for each phoneme frame 13 MFCC features are extracted. After feature extraction, multi-class labels are assigned for each phoneme according to the Tamil phoneme hierarchy as shown in figure 1. The dataset covers 37 Tamil phonemes and consists of 100 instances for each phoneme. There are totally 3700 instances where each instance has 45 class labels.

Multi-label classification task has been performed on the Tamil Phoneme dataset using the tool MEKA 1.9.0 [15][16]. Problem transformation methods such as Binary Relevance (BR), Label Power Set (LP), and Classifier Chain (CC) are used to build models and their performance is evaluated using the measures like accuracy, hamming-loss, one error, average precision.

The performance of the multi-label classifiers with different base classifiers that include Decision tree (J48), Naïve Bayes, SMO, AdaBoostM1 and ZeroR are presented in table 1 – 3.

Table 1. Performance of the Multi-label Tamil Phoneme classifier using BR

ML Classifier - BR	**Base Classifier**				
Metrics	**J48**	**NB**	**SMO**	**AdaBoostM1**	**ZeroR**
Accuracy	0.751	0.525	**0.797**	0.697	0.503
Hamming loss	0.031	0.076	**0.022**	0.038	0.082
One error	0.121	**0.115**	0.118	0.022	0.021
Avg precision	0.136	0.129	0.139	0.124	**0.169**

Table 2. Performance of the Multi-label Tamil Phoneme classifier using LC

ML Classifier - LC	**Base Classifier**				
Metrics	**J48**	**NB**	**SMO**	**AdaBoostM1**	**ZeroR**
Accuracy	0.775	0.892	**0.936**	0.473	0.467
Hamming loss	0.032	0.015	**0.009**	0.089	0.09
One error	0.091	0.055	**0.037**	0.343	0.343
Avg precision	**0.139**	0.135	0.134	0.137	0.137

Table 3. Performance of the Multi-label Tamil Phoneme classifier using CC

ML Classifier - CC	Base Classifier				
Metrics	**J48**	**NB**	**SMO**	**AdaBoostM1**	**ZeroR**
Accuracy	0.771	0.536	**0.821**	0.667	0.529
Hamming loss	0.031	0.094	**0.026**	0.042	0.06
One error	0.109	0.327	**0.107**	0.116	0.343
Avg precision	0.141	**0.162**	0.139	0.145	0.137

The results indicate that Label Power set problem transformation method using SMO base classifier provides better phoneme classification than other methods. The comparative analysis of the multi-label classifiers based on different performance measures are presented in figure 2-5.

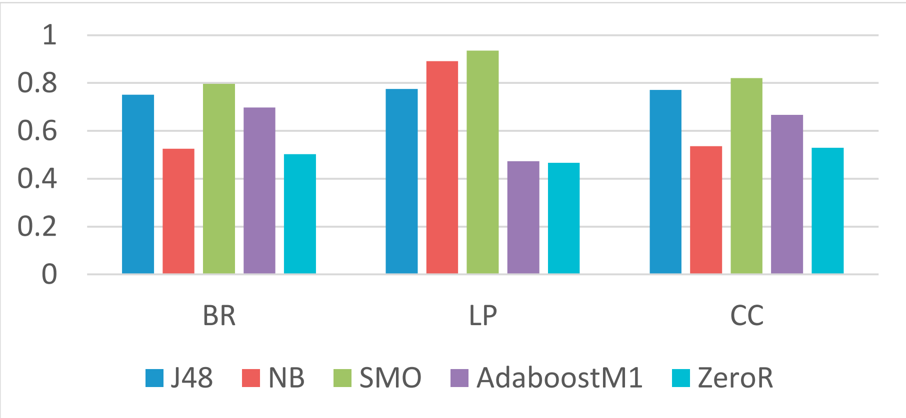

Fig. 2. Accuracy of classifiers with different base classifiers

The accuracy of the ML Classifier based on LP method with SMO classifier outperforms other classifiers.

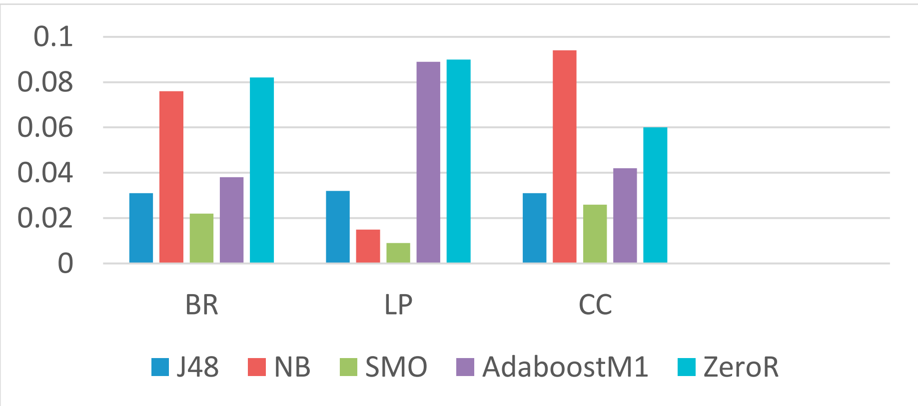

Fig. 3. Hamming- Loss of classifiers with different base classifiers

The hamming loss should be less or equal to zero for the best classifier. From fig.3, it is observed that LP based model with SMO base classifier is showing better performance which has less value for hamming loss.

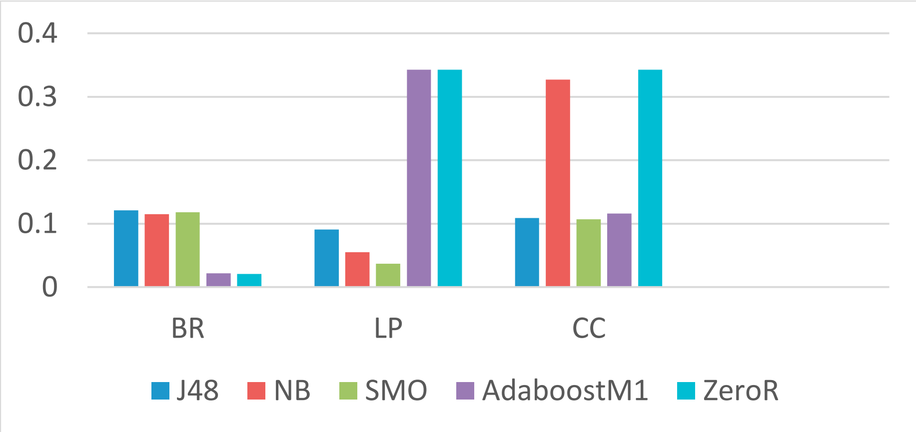

Fig. 4. One-Error classifiers with different base classifiers

Similar to hamming loss metric, one-error must be small for efficient classifier, according to fig. 4, both BR and LP based classifier are better.

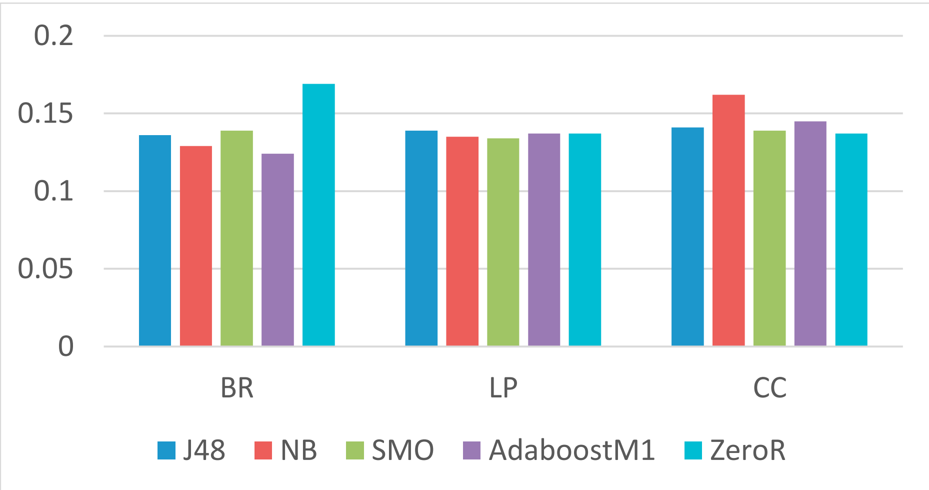

Fig. 5. Average Precision of classifiers with different base classifiers

The overall results indicate that in terms of accuracy LP based classifier outperforms other classifiers as well as in other metrics also the LP classifier with base classifier SMO provides significantly greater performance. Hence the multi-label classification can be very well employed in phoneme classification task in speech recognition.

6. Conclusion

Tamil language has approximately 45 unique phonemes. The phoneme classification tasks are useful in many speech recognition applications. In this paper, an attempt is made to classify the phonemes of Tamil language using multi-label classification algorithms. The task has been carried out with different problem transformation based methods such as BR, LP and CC with variety of base classifiers. The performance of the classifiers is analyzed using performance metrics like accuracy, hamming loss, one error and average precision. It is observed from the results that the LP classifier with SMO base classifier provides significantly better performance compared to other classifiers. In future, the task of Tamil phoneme classification can be extended by using other ML algorithms to improve the performance.

References

[1]. Eva Gibaja, Sebastia N Ventura. A Tutorial On Multi-Label Learning. *ACM Computing Surveys March* 2010; **9 (4)**, Article 39.

[2]. Min-Ling Zhang and Zhi-Hua Zhou. A Review on Multi-Label Learning Algorithms. *IEEE Transactions on Knowledge and Data Engineering* 2014; **26 (8)**.

[3]. Ilakkuvanar S. *Tholkappiyam (in English)*. India: Kural Neri Publishing House; 1963.

[4]. Karpagavalli S, Chandra E. A Hierarchical Approach in Tamil Phoneme Classification using Support Vector Machine. *Indian Journal of Science and Technology*. December 2015; Vol **8 (35)**, DOI: 10.17485/ijst/2015/v8i35/80681.

[5]. Karunakaran K, Jeya *V. Mozhiyiyal (in Tamil).* India: Kavitha Pathippakam; 1997.

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

[7]. LiLi BaobaoChang, ShiZhao, LeiSha, XuSun, HoufengWang. Multi-label Text Categorization with Joint Learning Predictions-as-Features Method. *In Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* September 2015; Lisbon, Portugal, 17-21, 835–839.

[8]. Tomar D, Agarwal S. Multi-label Classifier for Emotion Recognition from Music. *In Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics Smart Innovation, Systems and Technologies* 2016; **(43)**, Springer New Delhi.

[9]. Xuchun Li, Lei Wang, Eric Sung. Multi-Label SVM Active Learning for Image Classification. *IEEE Image Processing ICIP '04*, 2004.

[10]. Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. HCP: A Flexible CNN Framework for Multi-label Image Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2015.

[11]. Ricardo Cerri, Renato R. O. da Silva, and Andr´e C. P. L. F. de Carvalho. Comparing Methods for Multilabel Classification of Proteins Using Machine Learning Techniques. *Advances in Bioinformatics and Computational Biology* July 2009; 109-120, 10.1007/978-3-642-03223-3_10.

[12]. Erica Akemi Tanaka, Sérgio Ricardo Nozawa , Alessandra Alaniz Macedo , José Augusto Baranauskas. A multi-label approach using binary relevance and decision trees applied to functional genomics. *Journal of Biomedical Informatics* 2015; **54**: 85–95.

[13]. J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. *Machine Learning and Knowledge Discovery in Databases* 2009; **(5782)**: 254–269.

[14]. G. Tsoumakas and I. Vlahavas. Random k-labelsets: An ensemble method for multilabel classification. *In Proceedings of the 18th European Conference on Machine Learning, ECML '07, (Berlin, Heidelberg)* Springer-Verlag 2007; 406–417.

[15]. Vaishali S. Tidake, Shirish S. Sane. Multi-label Learning with MEKA. *CSI Communications* Aug 2016; **40 (5)**: 33-37.

[16]. https://sourceforge.net/projects/meka/files/meka-1.9.0.