Enhancing Accuracy of Multilabel Classification by Extracting Hierarchies

Alexander Ulanov, German Sapozhnikov*, Nickolay Lyubomishchenko, Vladimir Polutin and Georgy Shevlyakov*

HP Labs Russia, Saint-Petersburg, Russia
Email: {alexander.ulanov, nickolay.lyubomishchenko, vladimir.polutin}@hp.com
**Department of Applied Mathematics, Saint-Petersburg State Polytechnic University, Saint-Petersburg, Russia*
Email: gsapozhnikov@gmail.com, georgy.shevlyakov@phmf.spbstu.ru

***Abstract*—A novel algorithm of extracting hierarchies with the maximal F-measure for improving multilabel classification performance, the PHOCS, builds Predicted Hierarchy Of ClassifierS. Nodes contain classifiers, and each intermediate node corresponds to a set of labels, and a leaf node to a single label. Any classifier in the extracted hierarchy deals with a considerably smaller set of labels as compared to the number L of labels, and with a more balanced training distribution. This leads to an improved classification performance. Our method has linear training and logarithmic testing complexity with respect to L. The experiment was conducted on 4 multilabel datasets and it has confirmed the effectiveness of the PHOCS algorithm.**

***Keywords*-multilabel classification, taxonomy generation, hierarchy extraction, text mining.**

I. Introduction

The notion of classification is very general and has many applications, for instance, in text processing, computer vision, in medical and biological sciences, etc. The goal of text classification is to assign an electronic document to one or more categories based on its contents.

The traditional single-label classification deals with a set of documents associated with a single label (class) λ from a set of disjoint labels L, $|L| > 1$. To solve this problem, conventional tools are used, for instance, the Naive Bayes and Support Vector Machine classifiers [1]. If $|L| \geq 2$, then the learning problem belongs to multilabel classification [2], [3].

In some classification problems, labels are associated with a hierarchical structure, and in this case the task resides in the area of hierarchical classification. If each document corresponds to more than one node of a hierarchical structure, then we deal with hierarchical multilabel classification instead of flat (non-hierarchical) multilabel classification.

Methods of multilabel classification can generally be Divided into the following: problem transformation methods [4] [5] and algorithm adaptation methods [3].

Problem transformation methods are the methods transforming a multilabel classification problem into a single-label one, for the solution of which any classifier can be used. An essential property of problem transformation methods is that they are algorithm independent. Algorithm adaptation methods are the methods that extend specific learning algorithms to handle multilabel data directly.

If labels have a hierarchical structure, both hierarchical and flat classification algorithms can be used. However, in hierarchical classification, a hierarchy of classifiers can be built with the help of a label hierarchy. It is an important question why a hierarchical classification may perform better than a flat one.

First, with hierarchical classification we solve the problem similar to that of the class imbalance effect typical for single-label classification [6].

Second, computational complexity of training a multilabel Classifier strongly depends on the number of labels. Besides simple algorithms (e.g., binary relevance) with linear complexity with respect to $|L|$, there are also more advanced methods having higher complexity. Computational complexity of hierarchical classification is improved along with the linear training and logarithmic testing complexity with respect to $|L|$.

The main contribution of this work is suggestion of a novel algorithm of extracting hierarchies with the maximal F-measure for improving multilabel classification accuracy. The algorithm is called the PHOCS (Predicted Hierarchy Of ClassifierS). The principal idea is to enhance the accuracy of classification by transforming an original flat multilabel classification task with a large set of labels L into a tree-shaped hierarchy of simpler multilabel classification tasks.

Here, we propose the following solution:

- automatic generation of hierarchies for classification through flat clustering [1];
- use of certain criteria optimizing the F-measure for predicting and extracting prospective hierarchies;
- implementation of the corresponding toolkit on the basis of the WEKA ML tools.

The remainder of the paper is organized as follows. Section II outlines the state of the art in the area of hierarchical multilabel classification. Section III describes the PHOCS algorithm. Section IV presents the obtained results, and in Section V some conclusions are drawn.

II. State of the Art

In text classification, most of the studies deal with flat Classification, when it is assumed that there are no relationships between categories. Hierarchical classification is generally represented by two methods, namely, the big-bang approach and the top-down level-based approach [7].

In the big-bang approach, a document is assigned to a class in one single step, whereas in the top-down level-based approach, classification is performed with classifiers built at each level of a hierarchy.

In the top-down level-based approach, a classification problem is decomposed into a smaller set of problems corresponding to hierarchical splits in a tree. Each of these sub problems can be solved much more accurately [11], [13]. Moreover, a greater accuracy is possible to achieve because classifiers can identify and ignore commonalities between subtopics of a specific class, and concentrate on those features that distinguish them [12]. This approach is used by most hierarchical classification methods due to its simplicity [7] – [11]. They utilize a well-known hierarchical (taxonomy) structure built by experts.

One of the obvious problems with the top-down approach is that misclassification at a higher level of a hierarchy may force a document to be wrongly routed before it can be classified at a lower level. Another problem is that sometimes there is no predefined hierarchy and one has first to build it. It is usually built from data or from data labels. In our research we address the latter problem, which seems to us less complex from computational point of view, since the number of labels is usually less than the number of data attributes.

There exist approaches emploing linear discriminant projection of categories for creating hierarchies based on their similarities: [14], [15]. They show that classification performance in this case is better as compared to the case with a flat one. There is also a range of methods aimed to reduce the complexity of training flat classifiers. Usually they partition data into two parts and create a two-level hierarchy, e.g. [16].

The HOMER method [17] constructs a Hierarchy Of Multi-label classifiERs. Each of them deals with a much smaller set of labels with respect to $|L|$, and with a more balanced example distribution. This leads to an improved predictive performance as well as to linear training and logarithmic testing complexity with respect to $|L|$. At the first step, the HOMER automatically organizes labels into a tree-shaped hierarchy. This is accomplished by recursively partitioning a set of labels into a number of nodes using the balance clustering algorithm. Then it builds one multilabel classifier at each node apart from the leaves. (In the PHOCS, we use the same concept of hierarchy and metalabels.)

Tsoumakas et al. [18] introduce the RAkEL classifier (RAndom k labELsets, k is the parameter specifying the size of labelsets) that outperforms some well-known multilabel classifiers. (We use their results as a baseline for our method.)

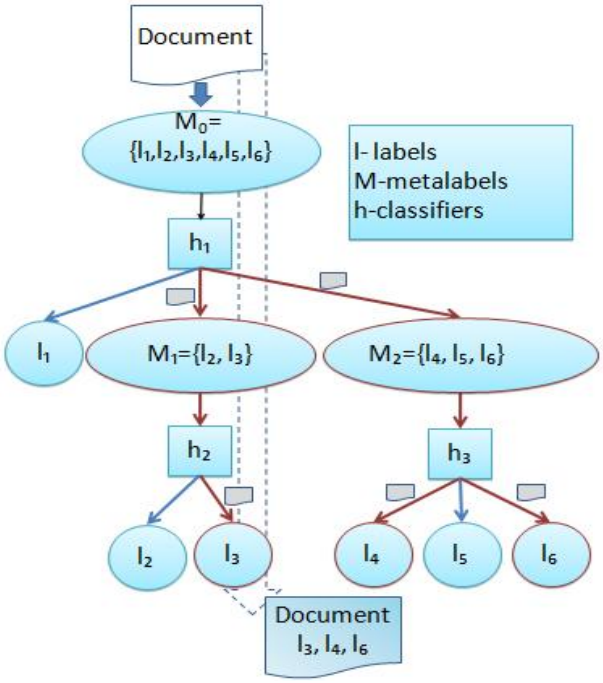

Figure 1. Hierarchical multilabel classification workflow

In the recent work [19], the authors use datasets with predefined hierarchies trying to guess, not to construct a hierarchy, which could be good for classification.

III. Algorithm For Building Hierarchies

A. General Concept

Following Tsoumakas et al. [3], we use the divide-and-conquer paradigm for the algorithm design. The main idea is transformation of a multilabel classification task with a large set of labels L into a tree-shaped hierarchy of simpler multilabel classification tasks, when each of them will deal with a small number k of labels: $k << |L|$ (sometimes $k < |L|$).

Each node n of this tree contains a set of labels $L_n \subseteq L$. In Figure 1 we have 6 leaves and 3 internal nodes. There are $|L|$ leaves, each containing a singleton (a single element set) $\{l_j\}$ with a different label j of L. Each internal node n contains a union of the label sets of its children: $L_n = \cup L_c$, $c \in children(n)$. The root accommodates all the labels: $L_{root} = L$.

We define a metalabel M_n of a node n as disjunction of the labels associated with that node: $M_n \vee l_j$, $l_j \in L_n$. Metalabels have the following semantics: a document is considered annotated with the metalabel M_n if it is annotated with at least one of the labels in L_n. Each internal node n of the hierarchy also accommodates a multilabel classifier h_n. The task of h_n is to predict one or more

metalabels of its children. Therefore, the set of labels for h_n is $M_n = \{M_c,\ c \in children(n)\}$. Figure 1 shows a sample hierarchy produced for a multilabel classification task with 6 labels.

For multilabel classification of a new document, the classifier starts with h_{root} and then forwards it to the multilabel classifier h_c of the child node c only if M_c is among the predictions of $h_{parent(c)}$. The main issue in building hierarchies is to determine the way of distributing the labels of L_n among the k children. One can distribute k subsets in such a way that labels belonging to the same subset will be similar. In [17] the number k of labels is given for each L_n. In this work we solve the problem of distributing labels L_n among children nodes by choosing the best value of k at each node according to the prediction of the hierarchy performance.

Algorithm 1 The PHOCS algorithm for hierarchy building

```
1: function HIERARCHY(TrainSet, Labels, RootNode)
2:     Pmin ← PerformanceMeasure(TrainSet)
3:     for i ← Kmin, Kmax do
4:         C[i] ← doClustering(TrainSet, Labels, i)
5:         DataSet ← dataMetaLabeling(TrainSet, C)
6:         Results[i] ← PerfMeasure(DataSet)
7:     end for
8:     PerfPredict, Kbest ← Predict(Results, C)
9:     if PerfPrediction > Pmin then
10:        addChildNodes(RootNode, C[Kbest])
11:        for i ← 0, BestNumber do
12:            Hierarchy(TrainSet,
13:            C[KBest][i], RootNode.Child(i))
14:        end for
15:    end if
16: end function
17: function PERFMEASURE(DataSet)
18:    TrainPart, TestPart ← split(DataSet)
19:    return Performance(TrainPart, TestPart)
20: end function
21: function PERFPREDICT(Results, Clusters)
22:    p[Kmin : Kmax] ← Results[Kmin : Kmax]
23:    p[Kmax + 1 : numOfLabels] = 0
24:    for i ← Kmax + 1, numOfLabels do
25:        for j ← 2, i do
26:            for k ← 2, i do
27:                p[i] ← Max(p[i], p[k] * p[j])
28:            end for
29:        end for
30:    end for
31:    for i ← Kmin, Kmax do
32:        p[i] ← Result[i] * maxClusterSize[i]
33:    end for
34:    return max(p), indexOfMax(p)
35: end function
```

B. The PHOCS Algorithm

A brief description of PHOCS is as follows (see Algorithm 1 and Figure 2). Our algorithm is recursive. It takes the following as an input: a training dataset, the minimum and maximum numbers of clusters k_{min} and k_{max} (line 1). It starts from the set of all labels, makes K-means clustering of them into different numbers of clusters from k_{min} to k_{max} (line 4). These sets of clusters are candidates for the next layer of the hierarchy. Each cluster contains a number of labels called a metalabel (line 5). Then we put them into the hierarchy and measure its classification efficiency. Thus we obtain the efficiency measure for each set (line 6). Next, we try to predict all options of the further development of each set (line 8). The best set is chosen and put into the hierarchy according to this prediction. The recursive process is performed until we receive clusters with single labels (line 12).

Having done partition of the labels into a certain number of clusters, we perform classification using these clusters. We use the $F1$-measure to measure how good the partition for classification is, particularly at a given layer. We want to predict the $F1$-measure for all possible hierarchy topologies using the results for the given layer and then to select the best one (line 21). The prediction for every topology depends on the $F1$-measure for the current layer and the number of labels still needed to be clustered. The question is in finding the best estimate of the $F1$-measure for the next layer of the hierarchy. It could be estimated by considering true positives, false negatives, and false positives (TP, FN, and FP), and then by computing the $F1$. We know that TP decreases and FN increases as we go deeper into a hierarchy. However, we cannot make any sound assumptions about FP (and true negatives as well). In this case they are difficult to predict, hence we use a simpler model. We assume that the $F1$-measure becomes smaller layer by layer (as a hierarchy is growing), that is, the $F1$-measure at the layer k is larger than at the layer $k+1$, and the relative decrease depends on the number of clusters. Finally, we estimate it as $F1_{k+1} = \prod_{i=1}^{k} F1_i$ (lines 24-30).

This multiplication formula can be explained as follows. If we dealt with the accuracy of classification assuming the errors at different layers to be independent of each other, then the accuracy at the $k+1$ layer would be found by multiplying the accuracies of all the previous layers. In our case we use the $F1$-measure which is not exactly the measure of the accuracy of classification but, nevertheless, it is close to it. Hence, the multiplication formula yields an approximation of the hierarchy performance. Thus, the final prediction for each partition depends both on the decrease of the $F1$-measure at the given layer and on the number of labels yet to be clustered, that is, on the size of the maximal cluster or, in other words, the maximal depth of the hierarchy. The final prediction can be found by multiplying

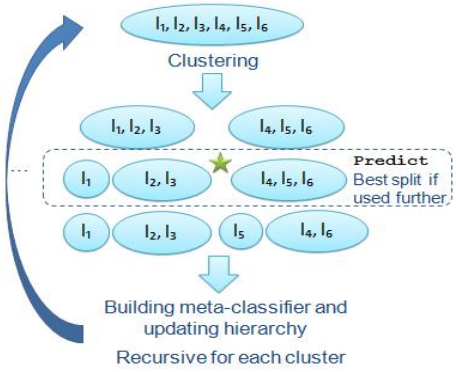

Figure 2. One step of predictive algorithm

the prediction for the maximal cluster and the initial results of classification over all clusters (lines 31-33). We have to notice that prediction and selection of the best topology could be made on the basis of other performance measures, like precision or recall (line 17). In this work we demonstrate the results only for the $F1$ measure.

IV. Experiment

A. Experiment Setup

The experiments are performed on 4 multilabel datasets available at http://mlkd.csd.auth.gr/multilabel.html. Table I presents basic statistics, such as the number of the examples and labels, along with the statistics that are relevant to the labelsets, such as upper bound of labelset, actual number, and diversity [18].

Table I
Multilabel Datasets and Their Statistics

Name	Examples	Labels	Bound of Labelsets	Actual Labelsets	Labelset Diversity
Medical	978	45	978	94	10%
Enron	1702	53	1702	753	44%
Mediamill	43907	101	43907	6555	15%
Bibtex	7395	159	7395	2856	39%

These datasets are divided into the train and test parts in the proportion 2 to 1, respectively. We do not do any other transformations of the datasets, in particular, not any attribute selection. The decision tree algorithm $C4.5$ [20] is chosen to be the basic multilabel classifier. For clustering and building hierarchies, the K-means algorithm is used. The micro and macro measures of classification accuracy (precision (P), recall (R) and $F1$-measure) are used in the same way as in [1]. The parameter values for the PHOCS are chosen as $k_{min} = 2$ and $k_{max} = 10$. Such k_{max} is chosen since the number of labels in our experiments has the order of 100, so the hierarchy contains at least 2 layers. Our algorithm was restricted to work on the first two layers, subsequent layers were created using flat clustering with maximum number of clusters $k_{max} = 10$. We build three hierarchies for each dataset: the hierarchies extracted with the accuracy and the micro $F1$-measure are similar and are marked as $H1$, whereas the $H2$-hierarchies correspond to the macro $F1$-measure.

B. Experimental Results

In the scope of PHOCS, there are only the datasets having no predefined hierarchies, thus the flat classification case is taken as the baseline. The obtained results for the flat and generated hierarchies are shown in Tables II - V. The significantly best values of the $F1$-measures are boldfaced. We are more interested in the results at the leaf nodes (labels), since the other labels are the metalabels not being present in the initial "flat" hierarchy.

Table II
Measures of Classification for the Mediamill Datasets: F1-measure, P - Precision, R - recall

	MICRO All Labels			MICRO Leaf Labels			MACRO Leaf Labels		
Hierarchy	F1	P	R	F1	P	R	F1	P	R
Flat	.54	.66	.45	.54	.66	.45	.10	.24	.08
H1	**.62**	.65	.58	.53	.58	.50	**.13**	.19	.11
H2	**.62**	.65	.58	.53	.58	.50	**.13**	.19	.11

Table III
Measures of Classification for the Bibtex Datasets

	MICRO All Labels			MICRO Leaf Labels			MACRO Leaf Labels		
Hierarchy	F1	P	R	F1	P	R	F1	P	R
Flat	.31	.81	.19	.31	.81	.19	.14	.40	.11
H1	**.61**	.73	.52	**.37**	.61	.21	**.22**	.38	.18
H2	**.57**	.66	.50	**.38**	.60	.27	**.22**	.40	.18

Table IV
Measures of Classification for the Medical Datasets

	MICRO All Labels			MICRO Leaf Labels			MACRO Leaf Labels		
Hierarchy	F1	P	R	F1	P	R	F1	P	R
Flat	.80	.85	.75	.80	.85	.75	.26	.32	.25
H1	**.88**	.91	.86	**.82**	.84	.81	**.30**	.33	.30
H2	**.88**	.91	.86	**.82**	.84	.81	**.30**	.33	.30

Table V
Measures of Classification for the Enron Datasets

	MICRO All Labels			MICRO Leaf Labels			MACRO Leaf Labels		
Hierarchy	F1	P	R	F1	P	R	F1	P	R
Flat	.46	.66	.35	.46	.66	.35	.09	.13	.08
H1	**.62**	.64	.61	**.50**	.62	.42	.10	.15	.09
H2	**.62**	.64	.60	.46	.59	.38	.10	.15	.08

The results for all the labels include the results at the intermediate metalabels. These results are represented only for micro measures, since in this case metalabels have a great impact, as a large number of documents pass them. In case of macro measures metalabels are less important, since their number is significantly smaller than that of leaf labels.

Next, we compare and analyze the results at the leaf labels. One can see in Tables II - V that with all datasets except the Mediamill the $F1$ measure of the extracted hierarchies outperforms the $F1$ measure of the flat one. It has been observed that the precision of classification slightly falls while the recall increases compared to the flat classification. This improves the $F1$-measure almost in all cases.

On average, in two cases out of four, the results are the same for the both measures used for prediction. Slight differences are observed on Bibtex and Enron datasets. These results show that sometimes we can adjust measures by extracting different hierarchies.

V. Conclusions

We propose a novel algorithm for automatic extraction of hierarchies for classification, called the PHOCS. This algorithm is based on flat clusterings of multilabels, and it is classifier independent. Thus, it has an advantage of enhancing the accuracy of multilabel classification by optimizing a hierarchy structure, not classifiers. The PHOCS is applicable for the datasets without predefined hierarchies. The experimental study of the PHOCS performance on 4 multilabel datasets proves its effectiveness. Implementation of the corresponding toolkit is made on the basis of the WEKA ML tools.

In the future we plan to use other criteria for performance prediction, like precision or recall. Currently we are carrying out experiments with agglomerative clustering instead of the flat one that could reduce the complexity of the PHOCS. We plan to explore with what kinds of tasks our algorithm can work better and why.

References

[1] C. D. Manning, P. Raghavan, and H. Schutze. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008.

[2] G. Xue, D. Xing, Q. Yang, and Y. Yu. Deep Classification in Large-scale Text Hierarchies. In: *Proc. 31st Annual International ACM SIGIR Conference on Research & Development on Information Retrieval(SIGIR 2008)*, July 20-24, 2008, Singapore.

[3] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining Multi-label Data. *Data Mining and Knowledge Discovery Handbook*, O. Maimon, L. Rokach (Ed.), Springer, 2nd edition, 2010.

[4] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, Vol. 37, No. 9, 2004, pp. 1757-1771.

[5] J. Read. A pruned problem transformation method for multi-label classification. In: *Proc. 2008 New Zealand Computer Science Research Student Conference (NZCSRS 2008)*, 2008, pp. 143-150.

[6] N. V. Chawla, N. Japkowicz, and A. Kotcz. Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explorations*, Vol. 6, 2004, pp. 1-6.

[7] A. Sun and E.-P. Lim. Hierarchical Text Classification and Evaluation. *Proc. IEEE Int'l Conf.Data Mining (ICDM'01)*, Nov. 2001, pp. 521-528.

[8] D. Koller and M. Sahami. Hierarchically Classifying Documents Using Very FewWords. In: *Proc. Intl Conf. Machine Learning (ICML 97)*, 1997, pp. 170-178.

[9] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced Hypertext Categorization Using Hyperlinks. In: *Proc. ACM SIGMOD 98*, 1998, pp. 307-318.

[10] A. S. Weigend, E. D. Wiener, and J. O. Pedersen. Exploiting Hierarchy in Text Categorization. *Information Retrieval*, Vol. 1, No. 3, 1999, pp. 193-216.

[11] S. T. Dumais and H. Chen. Hierarchical Classification of Web Content. In: *Proc. ACM SIGIR 2000*, 2000, pp. 256-263.

[12] S. D Alessio, K. Murray, R. Schiaffino, and A. Kershenbaum. The Effect of Using Hierarchical Classifiers in Text Categorization. In: *Proc. of RIAO-00, 6th International Conference "Recherche d'Information Assistee par Ordinateur"*, 2000, pp. 302-313.

[13] A. Pulijala and S. Gauch. Hierachcal Text Classification. In: Proc. CITSA 2004, 2004, Florida, USA, pp. 257-262.

[14] B. Gao, T. Y. Liu, Q. S. Cheng, G. Feng, T. Qin, and W. Y. Ma. Hierarchical Taxonomy Preparation for Text Categorization Using Consistent Bipartite Spectral Graph Copartitioning, *IEEE Transactions on Knowledge and Data Engineering (TKDE),* Vol. 17, 2005, pp. 1263-1273.

[15] T. Li, S.Zhu, and M. Ogihara. Hierarchical Document Classification Using Automatically Generated Hierarchy. *Journal of Intelligent Information Systems,* Vol. 29, 2007, pp. 211-230.

[16] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni Hierarchical classification: combining Bayes with SVM *Proceedings of the 23rd International Conference on Machine Learning,* 2006.

[17] G. Tsoumakas, I. Katakis, and I. Vlahavas. Effective and effcient multilabel classifcation in domains with large number of labels. In: *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, 2008, pp. 30-44.

[18] G. Tsoumakas, I. Katakis, and I. Vlahavas. Random k-Labelsets for Multi-Label Classification. *IEEE Transactions on Knowledge and Data Engineering,* Vol. 99, 2010.

[19] F. Brucker, F. Benites, and E. Sapozhnikova. Multi-label classification and extracting predicted class hierarchies. *Pattern Recognition*, Vol. 44, 2011, pp. 724-738.

[20] G. I. Webb. Decision Tree Grafting From the All-Tests-But-One Partition. In: *Proc. Sixteenth Int. Joint Conf. on Artificial Intelligence,* San francisco, CA, 1999, pp. 702-707.