A Threshold Based Multi-Label Classification

Durga Prasad Muni, Bintu G. Vasudevan and Rajesh Balakrishnan
Infosys Labs
Infosys Limited
Bangalore, India
e-mail: {DurgaPrasad_Muni, Bintu_Vasudevan, RajeshB}@infosys.com

***Abstract*—In classification problems, a pattern may belong to one or multiple categories. It is essential to deal multi-label classification accurately and efficiently. Threshold strategies can be used for multi-label classification. We propose four schemes to compute threshold for a threshold based multi-label classification. We validate our method using multi-label text data and multi-label image data.**

Keywords-multi-label, threshold, classification, data mining

I. INTRODUCTION

In a typical pattern classification problem, a single label/class is assigned to a given pattern from a pre-defined set of labels or classes. However, in many classification problems such as text categorization, image classification, music classification and medical diagnosis, a pattern (text/image/music/patient) may belong to more than one class or category simultaneously. For example, a news article describing a person who is both a player and a politician can be classified into two categories *Sports* and *Politics*. In multi-label classification, we assign one or multiple labels to a given pattern from a pre-defined set of labels or categories.

An overview of multi-label classification is given in [1]. The multi-label classification problem can be considered as a set of binary classification problems. For each category, this approach constructs a classifier by using data associated with this category as positive and all others as negative. For a given pattern, a probabilistic score indicative of the membership to the category is expected from the corresponding classifier. The threshold strategies can be applied to the set of scores to predict categories of a given pattern.

Yang [2] presented various threshold strategies and compares their effectiveness. The author grouped the threshold strategies into 3 groups: rank-based cut, proportion-based cut and score-based cut. Fan and Lin investigated selection of thresholds for score-based multi-label classification in [3]. The authors tuned the decision thresholds of the binary classifiers. Tang et. al. [4] proposed an approach called *Metalabeler*. It determined the relevant set of labels by learning the expected number of labels.

Sanden and Zhang [5] proposed a set of ensemble techniques specific to multi-label music genre classification. Wicker et. al. [6] proposed a multi-label classifier based on boolean matrix decomposition. Boolean matrix decomposition was used to extract latent labels representing useful boolean combinations of the original labels from the full label matrix. Cerri et. al. [7] used genetic algorithms (GA) for hierarchical multi-label classification. GA evolved the antecedents of classification rules. The set of evolved antecedents is selected to build the corresponding consequent of the rules. In [8], voting based learning classifier system was proposed for multi-label classification. In [9], a hypergraph spectral learning formulation was proposed for multi-label classification, where the hypergraph exploits correlation among class labels. To reduce computational cost, the authors proposed an approximate formulation that is equivalent to a least squares problem. To exploit dependencies between labels, Ghamrawi and McCallum [10] proposed multi-label conditional random field classification models that parameterize label co-occurrences in multi-label classification. Wang et. al. [11] proposed random-walk model based multi-label classification system. It mapped the multi-label patterns to graphs, on which the random walk was applied. For a given unlabeled pattern, the system transformed the original multi-label problem to some single-label sub-problems. Hariharan et. al. [12] proposed a max-margin formulation for the multi-label classification problem. The authors assumed labels are correlated but does not incorporate pairwise label terms in the prediction function. They also developed efficient optimization algorithms that are orders of magnitude faster than existing (cutting plane) methods.

In this paper, we present four schemes to determine threshold for multi-label classification. In section II, we give a brief introduction to multi-label classification with respect to threshold approach. In section III, we present our approach of threshold based multi-label classification. In section IV, we provide the experimental results and in section V we conclude.

II. MULTI-LABEL CLASSIFICATION

Let $X = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}\}$ be the given set of patterns. N is the size of the data set. Let $\{C_1, C_2, \ldots, C_L\}$ be the pre-defined categories. L is the total number of categories. A pattern $\mathbf{x_i}$ may belong to one or more categories. Let a set of

labels, S_i, be associated to $\mathbf{x_i}$, where $|S_i| \geq 1$. The data set D = $\{(\mathbf{x_i}, S_i), \forall\, i \in \{1,2,\ldots,N\}\}$. Suppose $\mathbf{x_2} \in C_1$, $\mathbf{x_2} \in C_3$ and $\mathbf{x_2} \in C_4$. Then the set of labels $S_2 = \{C_1, C_3, C_4\}$ is assigned to $\mathbf{x_2}$.

The multi-label classification problem can be considered together as a set of binary classification problems. In this approach, for each class C_j, a data set D_j is prepared from D. In D_j, all patterns $\{\mathbf{x_i}\} \in C_j$ are included with class label 1, i.e. the pairs $\{\mathbf{x_i}, 1\}$ are included to D_j and all patterns $\{\mathbf{x_i}\} \notin C_j$ are included with class label 0 (or -1). For class C_j, a classifier F_j is learned using the data set D_j. It is assumed that the classifier C_j provides a score f_j for a pattern $\mathbf{x}$ indicating the class membership of $\mathbf{x}$ to class C_j. This set of scores $\{f_1, f_2,\ldots, f_L\}$ is used to determine the set of labels (S_i) of the given pattern $\mathbf{x}$. Threshold strategies can be applied on this set of scores to predict the class labels of $\mathbf{x}$. In [2], the threshold strategies are described into three categories:

Rank-based Cut (RCut): It sorts scores for each pattern $\mathbf{x}$ and ranks the categories. Then the top *k* categories are assigned to $\mathbf{x}$. Usually, *k* is defined as the average length of class labels in the training data [4]. If average length is in between 2 and 3 then *k* can be taken either 2 or 3.

Proportion-based Cut (PCut): Here, for each category C_j, the test patterns are sorted by the scores for C_j and the class C_j is assigned to the top *k* patterns. The *k* is defined based on the prior probability of C_j estimated on training data. Since, test data as a batch is needed, so in real-world applications it is rarely used.

Score-based local optimization (SCut): It tunes the threshold for each category using a validation set. SCut optimizes the performance of the classifier on individual categories without guaranteeing a global optimum. This method is studied by Fan and Lin [3].

We have compared our approach with baseline approaches [4]. These approaches are:

a) Vanilla SVM (SVM_v): It is one-vs-Rest SVM without any post-processing technique. At the time of prediction, all the labels with a positive score are selected.

b) RCut: We have used RCut with *k* equal to average number of labels per pattern. The nearest integer of this real average number value is taken as either:

i) $k = \lfloor \text{Average Length} \rfloor$. This RCut is denoted as $RCut_c$ (RCut conservative).

ii) $k = \lceil \text{Average Length} \rceil$. This RCut is denoted as $RCut_a$ (RCut aggressive).

c) SCut: SCut tuned based on Micro-F1 is denoted as $SCut_i$ and SCut tuned based on Macro-F1 is denoted as $SCut_a$.

III. MULTI-LABEL CLASSIFICATION: OUR APPROACH

For a given pattern, we assign class C_j to the pattern if score $f_j \geq \theta_{avg}$, a threshold value. The value of θ_{avg} is computed using validation sets. We partition the training set D into a training set (D_{tr}) and a validation set (D_{val}) using 5-fold cross validation. We design a set of L binary classifiers using D_{tr} as mentioned in the previous section. We use support vector machine (SVM) as the classifier. We apply the classifiers on the validation set. For each instance $\mathbf{x}_i$ in the validation set, we obtain a set of probabilistic scores $\{f_1, f_2,\ldots, f_L\}$ corresponding to categories $\{C_1, C_2,\ldots, C_L\}$.

We find a value θ_i for each instance $\mathbf{x}_i$ of validation set. Then we take the average value of θ_i over all instances in the validation set as θ. After computing θ for each validation set of 5-fold cross validation, we take the average of the five θ values to obtain threshold value θ_{avg}.

We propose four approaches to compute θ_i.

1) Approach1: In the first approach, we sort the scores $\{f_1, f_2,\ldots, f_L\}$ of validation instance $\mathbf{x}_i$ in descending order. If *k* is the total number of labels are associated to $\mathbf{x}_i$ then we take the k^{th} score of the sorted scores of $\mathbf{x}_i$ **as** θ_i.

2) Approach2: In the second approach, θ_i is the minimum of the *k* scores corresponding to the *k* labels that are associated to $\mathbf{x}_i$.

3) Approach3: In the third approach, if *k* is the total number of labels are associated to $\mathbf{x}_i$ then we consider $(k+d)^{th}$ score after sorting the scores in descending order. Also, we consider the minimum of the *k* scores corresponding to the *k* labels associated to $\mathbf{x}_i$. The larger of these two scores is taken as θ_i. For RCV1 data, we have taken $d = 2$ and for scene data we have chosen the value of *d* as 1 (as number of total categories is small).

4) Approach4: In the fourth approach, if *k* is the total number of labels are associated to $\mathbf{x}_i$ then we consider the average of the k^{th} score (after sorting) and the minimum of *k* scores corresponding to the *k* labels associated to $\mathbf{x}_i$ as θ_i.

After obtaining θ_{avg} for above mentioned approaches, we train SVM with the complete training set D and then test the classifiers on the test data. While classifying test samples, we use corresponding θ_{avg} of the above mentioned approaches to determine classes.

IV. EXPERIMENTAL RESULTS

We used LIBSVM [13] tool to validate our approach. We implemented the 5-fold cross validation to optimize the parameters of SVM.

A. Data Sets

To validate our approach, we used the benchmark multi-label RCV1 five subset data sets [14] and multi-label scene classification data [15].

RCV1 (Reuters Corpus Volume I) Data: It is a text data (news documents). It has five subsets. Each subset has 3000 data points for training and 3000 data points for test, with in total 103 categories (topics). Most instances are labeled with multiple labels. Two categories in the 5 training sets do not contain any instances. So, we have removed these two categories from test sets. After dropping these two classes, it contained 101 categories. The instances are represented by 47,236 features. The data sets are highly imbalanced.

Scene Data: This is an image data. The task is to recognize which of six possible scenes available in the given set of images. These scenes are beach, sunset, field, fall foliage, mountain and urban. The data set contains 1211 pictures for training and 1196 pictures for testing. The

pictures are represented by 294 attributes. Few data points are labeled with multiple labels. That means, few images contain more than one scene.

In [16], it is mentioned that for text data, linear SVM performs better. So, we used linear SVM for learning the classifiers from RCV1 text data. However, for scene data, we used Radial Basis Function (RBF) SVM.

B. Evaluation Measures

We adopted the measures mentioned in [3] for multi-label classification performance measure. These measures are exact match ratio, Macro-F1 and Micro-F1. These measures are defined below. Let M be the total number of test patterns. Let $\mathbf{y_i}$, $\hat{y}_i \in \{0,1\}^L$ be the actual label set and the predicted label set for pattern $\mathbf{x_i}$ respectively.

1) Exact Match Ratio:

$$\text{Exact Match Ratio} = \frac{1}{M}\sum_{i=1}^{M} I[y_i = \hat{y}_i]$$

I is the indicator function. $I[z] = 1$, if z is true and 0 otherwise. Exact match ratio is the extension of the accuracy for traditional classification. It does not consider partial match between the actual labels and prediction labels. Macro-F1 and Micro-F1 consider partial matches.

2) Macro F1:

$$\text{Macro-F1} = \frac{1}{L}\sum_{l=1}^{L} F_1^l$$

F_1^l is the F1 measure of C_l category. F1 measure is the harmonic mean of precision and recall.

$$F_1^l = \frac{2\sum_{i=1}^{M} y_i^l \hat{y}_i^l}{\sum_{i=1}^{M} y_i^l + \sum_{i=1}^{M} \hat{y}_i^l}$$

3) Micro F1:

$$\text{Micro-F1} = \frac{2\sum_{l=1}^{L}\sum_{i=1}^{M} y_i^l \hat{y}_i^l}{\sum_{l=1}^{L}\sum_{i=1}^{M} y_i^l + \sum_{l=1}^{L}\sum_{i=1}^{M} \hat{y}_i^l}$$

C. Results

Tables I-V show the results using our approaches and using baseline methods for RCV1 subset data sets. For baseline methods, we have taken the result given in [4]. For all RCV1 subset data sets, our threshold based approaches are giving better Macro-F1 as compared to the baseline methods. *Approach2* performs consistently well.

TABLE I. RESULT WITH RCV1 SUBSET 1 DATA

Approach	Measures		
	Exact	***Macro-F1***	***Micro-F1***
Approach1	31.97	44.15	65.89
Approach2	32.3	**45.28**	66.72
Approach3	32.13	44.82	66.43
Approach4	32.27	44.7	66.36
SVM_v	**39.93**	35.01	**72.88**
$RCut_c$	29.23	39.31	72.58
$RCut_a$	3.13	41.79	69.15
$SCut_i$	14.03	33.06	61.03
$SCut_a$	14.97	39.07	63.53

TABLE II. RESULT WITH RCV1 SUBSET 2 DATA

Approach	Measures		
	Exact	***Macro-F1***	***Micro-F1***
Approach1	38.17	43.79	73.43
Approach2	37.9	**45.26**	**74.43**
Approach3	38.27	45.02	74.27
Approach4	38.23	44.78	74.06
SVM_v	**40.07**	35.75	73.69
$RCut_c$	29.67	38.43	72.40
$RCut_a$	3.90	40.36	69.01
$SCut_i$	26.17	35.03	69.58
$SCut_a$	24.13	40.84	70.56

TABLE III. RESULT WITH RCV1 SUBSET 3 DATA

Approach	Measures		
	Exact	***Macro-F1***	***Micro-F1***
Approach1	31.9	41.86	65.28
Approach2	31.27	42.83	65.73
Approach3	31.43	42.41	65.51
Approach4	31.7	**43.17**	65.45
SVM_v	**41.67**	34.12	**73.91**
$RCut_c$	30.43	37.50	73.05
$RCut_a$	3.63	39.81	69.30
$SCut_i$	21.07	32.59	64.22
$SCut_a$	15.70	37.16	63.54

TABLE IV. RESULT WITH RCV1 SUBSET 4 DATA

Approach	Measures		
	Exact	***Macro-F1***	***Micro-F1***

Approach	Measures		
	Exact	*Macro-F1*	*Micro-F1*
Approach1	37.67	46.05	73.84
Approach2	37.5	**47.56**	**74.53**
Approach3	37.4	47.54	74.31
Approach4	37.57	46.08	74.08
SVM_v	**39.37**	33.04	72.98
$RCut_c$	29.43	40.16	72.81
$RCut_a$	3.47	43.19	69.52
$SCut_i$	20.47	32.86	66.19
$SCut_a$	20.93	40.18	68.81

TABLE V. RESULT WITH RCV1 SUBSET 5 DATA

Approach	Measures		
	Exact	*Macro-F1*	*Micro-F1*
Approach1	32.3	42.72	63.99
Approach2	31.9	**44.31**	64.99
Approach3	32.13	43.83	64.73
Approach4	32.23	44	64.45
SVM_v	**38.10**	34.47	**72.84**
$RCut_c$	28.93	37.46	71.83
$RCut_a$	2.97	39.62	68.27
$SCut_i$	33.60	33.70	67.61
$SCut_a$	33.90	41.50	70.57

We provided the result for scene data in Table VI. Here, we have compared our method (with various schemes to compute threshold) against our implementation of $RCut_c$ method. We obtained marginally better result for all 3 measures.

Our approaches provided consistently better Macro-F1. By definition, Macro-F1 is more sensitive to the performance of rare categories and Micro-F1 is more influenced by the major categories [4]. Since we obtained high Macro-F1 value, hence our approaches could be suitable for rare categories.

We also tried multi-class approach (multi-class SVM) for the multi-label classification and used these three measures to compute performance. We observed that binary approach is giving better result than multi-class approach, so we didn't include the multi-class approach results.

TABLE VI. RESULT WITH SCENE DATA

Approach	Measures		
	Exact	*Macro-F1*	*Micro-F1*
Approach1	**59.85**	**54.97**	64.19
Approach2	59.35	54.91	**64.24**
Approach3	59.35	54.86	64.18
Approach4	59.52	54.88	64.15
$RCut_c$	57.86	52.82	61.92

V. CONCLUSION

We proposed four approaches to compute threshold values for a threshold based multi-label classification. We used SVM to learn the classifiers from data. We decomposed the multi-label classification problem into a set of binary classification problems and then used threshold approach to predict the class labels. We used 5-fold cross validation to compute threshold value for classification and to find the value of parameters of the SVM. We validated our approach using benchmark RCV1 multi-label data sets and multi-label Scene classification data set. We obtained consistently better Macro-F1 against baseline approaches. This indicates that our method could capture the rare categories well.

REFERENCES

[1] G. Tsoumakas and K. Ioannis, "Multi-label classification: An overview," International Journal of Data Warehousing and Mining," vol. 3, 2007, pp. 1-13.

[2] Y. Yang, "A study of thresholding strategies for text categorization," Proceedings of the 24th research and development in information retrieval (SIGIR 01), 2001, pp. 137-145.

[3] R.-E. Fan and C.J. Lin. "A study on Threshold Selection for multi-label Classification," 2007.

[4] L. Tang, S. Rajan, and V. K. Narayanan,"Large scale multi-label classification via metalabeler," Proceedings of the 18th international conference on World Wide Web (WWW 09), 2009.

[5] C. Sanden and J. Z. Zhang, "Enhancing multi-label music genre classification through ensemble techniques," Proceedings of the 34th international ACM SIGIR conference on research and development in information retrieval (SIGIR 11), July 2011, pp. 705 – 714.

[6] J. Wicker, B. Pfahringer and S. Kramer, "Multi-label classification using Boolean matrix decomposition," Proceedings of the 27th annual ACM Symposium on Applied Computing (SAC 12), march 2012, pp. 179 – 186.

[7] R. Cerri, R. C. Barros and A.C.P.L.F. de Carvalho, "A genetic algorithm for hierarchical multi-label classification," Proeedings of the 27th annual ACM Symposium on Applied Computing (SAC 12), march 2012, pp. 250 – 255.

[8] K. Ahmadi-Abhari, A. Hamzeh and S. Hashemi, "Voting based learning classifier system for multi-label classification," Proceedings of the 13th annual conference companion on genetic and evolutionary computation (GECCO 11), July 2011, pp. 355 – 359.

[9] L. Sun, S. Ji and J. Ye, "Hypergraph spectral learning for multi-label classification," Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data mining (KDD 08), August 2008, pp. 668 – 676.

[10] N. Ghamrawi and A. McCallum, "Collective multi-label classification," Proceedings of the 14th ACM international Conference on Information and Knowledge Management (CIKM 05), October 2005, pp. 195 – 200.

[11] C. Wang, W. Zheng, Z. Liu, Y. Bai and J. Wang, "Using random walks for multi-label classifcation," Proeedings of the 20th ACM

international Conference on Information and Knowledge Management (CIKM 11), October 2011, pp. 2197 – 2200.

[12] B. Hariharan, L. Zelnik-Manor, S.V. N. Vishwanathan and M. Varma, "Large scale max-margin multi-label classification with priors," Proceedings of the 27th international Conference on Machine Learning (ICML 10), 2010.

[13] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," ACM Transactions on intelligent systems and technology, vol. 2, no. 27, 2011, pp. 1 – 27.

[14] D. D. Lewis, Y. Yang, T.G. Rose, and F. Li,"Rcv1: A new benchmark collection for text categorization research," Journal of machine Learning research, vol. 5, 2004, pp. 361-397.

[15] 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.

[16] T. Joachims, "Text Categorization with Support Vector Machines: Learning with many relevant features," Proceedings of 10th European Conference on Machine Learning (ECML 98), 1998, pp. 137-142.