First publl. in: Proceedings / Second International Workshop on Knowledge Discovery and Data Mining (WKDD 2009), 2009, pp. 144-147

Multi-label Classification with ART Neural Networks

Elena P. Sapozhnikova
Nycomed Chair for Applied Computer Science
University of Konstanz
Konstanz, Germany
elena.sapozhnikova@uni-konstanz.*de*

***Abstract*—Multi-label Classification (MC) is a classification task with instances labelled by multiple classes rather than just one. This task becomes increasingly important in such fields as gene function prediction or web-mining. Early approaches to MC were based on learning independent binary classifiers for each class and combining their outputs in order to obtain multi-label predictions. Alternatively, a classifier can be directly trained to predict a label set of an unknown size for each unseen instance. Recently, several direct multi-label learning algorithms have been proposed. This paper investigates a novel method to solve a MC task by using an Adaptive Resonance Theory (ART) neural network. A modified Fuzzy ARTMAP algorithm Multi-Label-FAM (ML-FAM) was applied to classification of multi-label data. The obtained preliminary results on the Yeast data set and their comparison with the results of existing algorithms demonstrate the effectiveness of the proposed approach.**

Keywords-Multi-label Classification, Neural Networks, Fuzzy ARTMAP

I. INTRODUCTION

Machine learning and neural network classifiers have been extensively studied in the past decades in the one-class-per-instance setting. However many real world problems produce more complex data sets, for example with classes that are not necessarily mutually exclusive. So, a gene can have multiple biological functions or a text document can be assigned to multiple topics. Thus, Multi-label Classification (MC) when an instance could belong to more than one class becomes an issue. Generally, a MC task is more difficult to solve than a single-label classification task. The main problem is a large number of possible class label combinations and the corresponding sparseness of available data. In addition, standard classifiers cannot be directly applied to a MC problem for two reasons. First, most standard algorithms assume mutually exclusive class labels, and second, standard performance measures are not suitable for evaluation of classifiers in a MC setting.

A traditional approach to MC is to learn multiple independent binary classifiers to separate one class from the others and then to combine their outputs. However in such a case the labels in the label set are treated as independent that can significantly reduce classification performance [1, 2]. Recently, several direct approaches [1-8] specially developed for solving a MC task have been proposed. Some of them are based on machine learning techniques like Support Vector Machine (SVM) [2-4], *k*-Nearest Neighbor (*k*NN) classification algorithm [5-7] or neural networks [8]. Although being able to achieve high accuracy, they all are conventional "black-box" classifiers that focus on predictive performance rather than knowledge extraction. Alternatively, an earlier attempt to use interpretable multi-label classifiers has been made in bioinformatics [9] in order to learn predictive rules from gene data by the decision tree algorithm C4.5 [10]. But the task was not the complete classification as the authors stated; they were mainly interested in obtaining a small set of accurate rules. A study of hierarchical MC with decision trees has been subsequently made in [11]. Two other generalizations of decision trees to MC are reported in [12] and [13]. However until now, there is still little prior work in using interpretable classifiers for MC. So, it has been no attempt to date to use hybrid neuro-fuzzy methods for MC.

In this paper, a multi-label extension of Fuzzy ARTMAP (FAM) [14] named Multi-Label FAM (ML-FAM), an interpretable classifier based on the Adaptive Resonance Theory (ART), is presented. FAM belongs to neuro-fuzzy hybrid algorithms which are well-suited for fuzzy rule extraction as opposite to traditional neural networks.

The paper is organized as follows. Section II presents a brief description of the FAM algorithm. In Section III, its key modifications especially developed for solving a MC task are introduced. Section IV describes experiments and compares obtained results with the results of some other classifiers reported earlier in the literature. And the final section concludes the paper by discussion and outlook.

II. FAM NEURAL NETWORK

The reader is supposed to be familiar with FAM. Due to space constraints only the basic steps of the algorithm are described. A FAM system generally consists of two self-organizing two-layer Fuzzy ART modules, ART_a and ART_b, which process input and target vectors respectively by building prototype categories in their second layers. These modules are linked by the Map Field – an associative memory, which forms associations between ART_a and ART_b prototype categories. During supervised learning, generation of categories in ART_a is guided by the target information coded in ART_b. In classification tasks, the target vectors usually represent class labels, for example in the binary form.

Konstanzer Online-Publikations-System (KOPS)
URN: http://nbn-resolving.de/urn:nbn:de:bsz:352-opus-87956
URL: http://kops.ub.uni-konstanz.de/volltexte/2009/8795/

The steps below are identical for both modules ART_a and ART_b. For simplicity, only ART_a operations are listed because distinguishing between the modules is not important here. Initially, the weight vectors are set to unity $w_{j1}(0)=\ldots=w_{jL}(0)=1$, for all prototype nodes $j=1,\ldots, N$ and each node is said to be uncommitted. The weights from ART_a to the Map Field w_{jk}^{ab} which store associations between committed nodes at ART_a and those at ART_b are initially also set to unity. The Map Field is connected to ART_b by one-to-one, non-adaptive pathways in both directions.

An M-dimensional input vector $\boldsymbol{a}=(a_1, a_2,\ldots, a_M)$ with the components a_i in the interval [0,1] is normalized by complement coding as $\boldsymbol{A}=(\boldsymbol{a}, \boldsymbol{a}^c) \equiv (a_1, \ldots, a_M, 1\text{-}a_1, \ldots, 1\text{-}a_M)$. Thereafter, the city-block norm L_1 denoted as $|\ldots|$ is equal for all input vectors: $|\boldsymbol{A}|= |(\boldsymbol{a}, \boldsymbol{a}^c)| =M$.

The choice function (1) is calculated for each node, and the best matching prototype J is found according to Winner-Take-All (WTA) rule (2):

$$T_j(\boldsymbol{A}) = \frac{|\boldsymbol{A} \wedge \boldsymbol{W}_j|}{\alpha + |\boldsymbol{W}_j|} \tag{1}$$

$$T_J = \max\{T_j : j = 1,...,N\} \tag{2}$$

where „$\wedge$“ denotes the fuzzy AND, element-wise min operator, and $\alpha>0$ is called the choice parameter. The network category choice J must be confirmed by checking the match criterion:

$$\frac{|\boldsymbol{A} \wedge \boldsymbol{W}_J|}{|\boldsymbol{A}|} \geq \rho_a \tag{3}$$

where $\rho_a \in [0,1]$ is the user-defined vigilance parameter. If it fails, the system inhibits the winning node J and enables another category to be selected.

During training, input vectors $\boldsymbol{a}$ are presented to the network together with their corresponding targets $\boldsymbol{b}$. After they have been processed in the respective modules, the Map Field vigilance criterion (4) tests the correctness of the association between the node J coding $\boldsymbol{a}$ and the node K coding $\boldsymbol{b}$

$$\frac{|\boldsymbol{U}^b \wedge \boldsymbol{W}_J^{ab}|}{|\boldsymbol{U}^b|} \geq \rho_{ab} \tag{4}$$

where $\boldsymbol{U}^b$ denotes the ART_b output vector and $\boldsymbol{W}_J^{ab}$ denotes the weights connecting the Jth ART_a node and the Map Field.

If the chosen ART_a category is disconfirmed by the ART_b output vector, then $\boldsymbol{U}^b \wedge \boldsymbol{W}_J^{ab} = 0$ and inequality (4) fails. In such a case the Match Tracking process initiates the choice of a new category by increasing the ART_a vigilance parameter ρ_a to the value slightly greater than the left-side term of (3). This search process continues until the input is either assigned to an existing (committed) node that satisfies both the ART_a match criterion (3) and the Map Field vigilance criterion (4) or to activation of a new (uncommitted) neuron.

A successful end of search leads to the learning changes in the weight vectors of the winning nodes at ART_a and ART_b:

$$\boldsymbol{W}_J^{(new)} = \beta(\boldsymbol{A} \wedge \boldsymbol{W}_J^{(old)}) + (1-\beta)\boldsymbol{W}_J^{(old)} \tag{5}$$

where $\beta \in [0,1]$ is the learning rate. The fast learning mode is achieved by setting $\beta=1$. The Map Field weights approach $\boldsymbol{U}^b \wedge \boldsymbol{W}_J^{ab}$ during training, and once J learns to predict K, that association is permanent, i.e. $w_{JK}^{ab}=1$ while $w_{Jk}^{ab}=0$ ($k \neq K$).

III. ML-FAM

Although ART networks like FAM enable multi-label learning, their direct use for solving a MC task can be ineffective due to the WTA choice rule (2). It allows only the most highly activated category to be selected. Though justified for mutually exclusive classes, this can lead to poor performance in the multi-label setting because only one set of labels can be predicted for an instance no matter how close to each other may be different label combinations. Thus, it would be advantageous to utilize distributed activation at ART_a during the classification stage in order to extract more information about dependencies and correlations between different label sets.

Distributed activation has been already shown to be effective in pattern recognition by ART networks [15]. However, it was used only for resolving ambiguity during category choice and accumulating evidence from multiple views in the single-label setting. In the case of MC, simultaneous activation of multiple categories does not necessarily mean an ambiguous prediction, but may be caused by correlations between labels. So, a better prediction can be made by joining the class information of those categories which are about equally activated at ART_a. This can be achieved by combining the individual predictions of several most activated nodes. The proposed method is implemented in ML-FAM as follows.

First, a set of N best categories with the largest activation values is chosen according to the following rule: a category j is included in the set, if the relative difference of activations $(T_{max}\text{-}T_j)/T_{max}$ lies below a predefined threshold t. Then the activations are normalized according to

$$u_j^a = \frac{T_j}{\sum_{n=1}^{N} T_n}, \tag{6}$$

and the resulting prediction $\boldsymbol{P}$ is made by calculating a weighted sum of N individual predictions $\boldsymbol{p}_j$

$$\boldsymbol{P} = \sum_{n=1}^{N} u_j^a \boldsymbol{p}_j \tag{7}$$

Thus, $\boldsymbol{P}$ contains a score for each label which is proportional to the frequency of predicting this label among N best categories. And finally, a post processing filter method outputs only those labels from $\boldsymbol{P}$ for which the obtained score is greater than a predefined fraction s of the highest score.

An addition modification is made in the ART_b weights which now count label frequencies for each ART_b category during learning. They are initially set to 0 and increased by 1 each time the corresponding label occurs. The weight matrix $\boldsymbol{W}^b$ then contains information about the frequency with which each label was coded by a node k. It should be noted that this modification causes some changes in the ART_b activation function of FAM which is now computed as $T_j = |\boldsymbol{B} \wedge \boldsymbol{W}_j^b|$. Another difference between FAM and ML-FAM is that the latter network does not use the Map Field vigilance parameter and Match Tracking with raising vigilance. The winner node is simply inhibited and a new search is started when the chosen ART_a category does not code the proper label set of a training instance.

The presented modifications can be also used with other ART-based fuzzy networks such as, for example, fuzzy ARAM [16].

IV. EXPERIMENTS

A. Data set

The effectiveness of standard FAM with the WTA choice rule and ML-FAM in the multi-label setting was evaluated on the well-known Yeast data set [2] describing the genes of *Saccharomyces cerevisiae*. Each gene is characterized by 103 features derived from the micro-array expression data and phylogenetic profiles. 14 possible classes in the top level of the functional hierarchy are considered with the average value of 4.24 labels per gene. For performance comparison, either the same splitting the data in 1500 training and 917 test instances as used in the literature [2, 5, 17] was adopted or a ten-fold cross-validation was performed as in [7, 8].

B. Performance measures

Performance evaluation of MC is different from that of usual single-label classification. The most common MC performance measures which are also used in this paper include *Hamming Loss* (*HL*), *Accuracy* (*A*), *Recall* (*R*) *Precision* (*P*), and *One-Error* (*OE*). While the first four metrics are defined on the basis of set operations [3], *OE* is based on ranking and evaluates how many times the top-ranked label is not in the true set of labels of the instance. Given a set $S=\{(\mathbf{x}_1, \mathbf{Y}_1), \ldots, (\mathbf{x}_n,\mathbf{Y}_n)\}$ of n test examples where $\mathbf{Y}_i$ is the proper label set for an instance $\mathbf{x}_i$, let $\mathbf{Z}_i$ be the set of predicted labels for $\mathbf{x}_i$ and $\mathbf{L}$ the finite set of possible labels. Then *HL* counts prediction errors when a false label is predicted as well as missing errors when a true label is not predicted:

$$HL = \frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{Y}_i \Delta \mathbf{Z}_i|}{|\mathbf{L}|} \quad (8)$$

where Δ denotes the symmetric difference of two sets and corresponds to the XOR operation in Boolean logic. The smaller is the *HL* value, the better is the MC performance.

Accuracy, *Recall* and *Precision* are defined as follows:

$$A = \frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Y}_i \cup \mathbf{Z}_i|} \quad (9)$$

$$R = \frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Y}_i|} \quad (10)$$

$$P = \frac{1}{n}\sum_{i=1}^{n}\frac{|\mathbf{Y}_i \cap \mathbf{Z}_i|}{|\mathbf{Z}_i|} \quad (11)$$

Given a real-valued function f which assigns larger values to labels in $\mathbf{Y}_i$ than those not in $\mathbf{Y}_i$, the *OE* metric can be defined as:

$$OE = \frac{1}{n}\sum_{i=1}^{n}\delta\left(\left[\arg\max f(\mathbf{x}_i, y)\right] \notin \mathbf{Y}_i\right) \quad (12)$$

where δ is a function that outputs 1 if its argument is true and 0 otherwise. The performance is perfect when *OE* equals to 0.

C. Classification results

Since the values of the vigilance parameters in ART_a and ART_b significantly affect classification performance and the network size, several experiments were made in order to find a proper setting. In FAM, ρ_a was set to 0 to minimize the network size and ρ_b was equal to 1. In ML-FAM, ρ_a and ρ_b were chosen as 0.7 and 1, and the parameters t and s as 0.05 and 0.6, respectively. Both networks were trained in the fast learning mode with β=1. The choice parameter was set to 0.0001.

Table I reports the average classification results of FAM and ML-FAM classifiers which were obtained on 10 independent runs. This is necessary to reduce the influence of the input presentation order which is significant under the fast learning condition. Note that *OE* is not a suitable performance measure for FAM because it outputs no ranked labels. The average number of created categories was 826 for FAM and 340 for ML-FAM. As expected, ML-FAM outperforms the single-label version on all performance measures and builds also a smaller network.

Table II shows comparative classification results of the standard single-label classifiers *k*NN and C4.5 on the Yeast data set from [17]. The columns correspond to two problem settings. In the first one, $|\mathbf{L}|$ independent binary classifiers were trained to separate one class from the others. In the second setting, each different set of labels in the data was considered as a single label for learning one single-label classifier. Table III presents the results obtained on the same data set by the multi-label classifiers BoosTexter, ML-*k*NN, and multi-label Alternating Decision Tree (ADT) taken from [5]. ML-FAM is superior to single-label classifiers as well as to multi-label classifiers BoosTexter and ADT.

TABLE I. CLASSIFICATION RESULTS OF FAM AND ML-FAM ON THE YEAST DATA (AVERAGED ON 10 RUNS)

Performance measure	FAM		ML-FAM	
	mean	*std*	*mean*	*std*
HL	0.231	0.007	0.205	0.002
A	0.477	0.019	0.517	0.004
R	0.571	0.028	0.602	0.005
P	0.629	0.014	0.692	0.003
OE	-	-	0.245	0.008

TABLE II. CLASSIFICATION RESULTS OF SINGLE-LABEL CLASSIFIERS ON THE YEAST DATA FROM [17]

Performance measure	|L| binary classifiers		One classifier	
	kNN	*C4.5*	*kNN*	*C4.5*
HL	0.243	0.259	0.229	0.286
A	0.479	0.423	0.495	0.399
R	0.601	0.593	0.628	0.528
P	0.596	0.561	0.596	0.529

TABLE III. CLASSIFICATION RESULTS OF MULTI-LABEL CLASSIFIERS ON THE YEAST DATA FROM [5]*

Performance measure	BoosTexter	ML-*k*NN *k*=8	ML-*k*NN *k*=9	ADT
HL	0.237	0.197	0.197	0.213
OE	0.302	0.248	0.251	0.245

**A*, *R* and *P* measures were not used in the reference.

Table IV compares the cross-validation results of FAM and ML-FAM with those reported for BoosTexter, Rank-SVM, ML-*k*NN, and ADT in [6], for DML-*k*NN in [7] and for BP-MLL in [8]. ML-FAM is superior to FAM, BoosTexter, ADT, and Rank-SVM; it is comparable to BP-MLL, but its performance is worse than that of ML-*k*NN and DML-*k*NN. However, it is worth noting that the *k*NN-based algorithms memorize all training examples as opposite to ML-FAM which creates only about 448 prototype categories in this experiment.

V. CONCLUSIONS

In this paper, a multi-label extension to Fuzzy ARTMAP named ML-FAM is presented. It can be successfully used for solving a multi-label classification task. The preliminary experiments show that the performance of the proposed classifier is superior to or comparable with the performance of other multi-label classifiers, except for multi-label *k*NN algorithms. This work should be continued by evaluating the method on other data sets used in the literature which will make the result comparison easier. Another point of the future work is replacing one-to-one Map Field mapping by one-to-many mapping.

ACKNOWLEDGMENT

This project is supported by the German Research Foundation (Deutsche Forschungsgemeinschaft (DFG), Bonn-Bad Godesberg, Germany.

REFERENCES

[1] R. E. Schapire, Y. Singer, "Boostexter: a boosting-based system for text categorization," Machine Learning, vol. 39, No. 2-3, 2000, pp. 135-168.

[2] A. Elisseeff, J. Weston, "A kernel method for multi-labeled classification," Advances in Neural Information Processing Systems, 2001, pp. 681-687.

[3] S. Godbole, S. Sarawagi, "Discriminative methods for multi-labeled classification," In LNCS, Advances in Knowledge Discovery and Data Mining, vol. 3056, No. 1, 2004, pp. 22-30.

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

[5] M.-L. Zhang, Z.-H. Zhou, "A k-nearest neighbor based algorithm for multi-label classification," Proc. International Conf. on Granular Computing, vol. 2, 2005, pp. 718-721.

[6] M.-L. Zhang, Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," Pattern Recognition, vol. 40, No.7, 2007, pp. 2038-3048.

[7] Z. Younes, F. Abdallah, and T. Denoeux, "Multi-label classification algorithm derived from k-nearest neighbor rule with label dependencies". Proc. 16th European Signal Processing Conf., 2008.

[8] 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, vol. 18, No. 10, 2006, pp. 1338-1351.

[9] A. Clare, R. D. King, "Knowledge discovery in multi-label phenotype data," Proc. 5th European Conf. on Principles of Data Mining and Knowledge Discovery, 2001, pp. 42-53.

[10] J. R. Quinlan, C4.5: Programs for Machine Learning, San Francisco: Morgan Kaufmann, 1993.

[11] H. Blockeel, M. Bruynooghe, S. Dzeroski, J. Ramon, J. Struyf, "Hierarchical multi-classification," Proc. First International Workshop on Multi-Relational Data Mining, 2002, pp. 21-35.

[12] E. Suzuki, M. Gotoh, and Y. Choki, "Bloomy Decision Tree for Multi-objective Classification," Proc. 5th European Conf. on Principles of Data Mining and Knowledge Discovery, 2001, pp. 436-447.

[13] F. De Comité, R. Gilleron, and M. Tommasi, "Learning multi-label alternating decision trees from texts and data," In LNCS, vol. 2734, 2003, pp. 251-274.

[14] G. A. Carpenter, S. Grossberg, N. Markuzon, J. H. Reynolds, and D. B. Rosen, "Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps," IEEE transactions on Neural Networks, vol. 3, 1992, pp. 698-713.

[15] G. A. Carpenter, W. D. Ross, "ART-EMAP: A neural network architecture for object recognition by evidence accumulation," IEEE Transactions on Neural Networks, vol. 6, 1995, pp. 805-818.

[16] A.-H. Tan, "Adaptive resonance associative map," Neural Networks, vol. 8, No. 3, 1995, pp. 437–446.

[17] G. Tsoumakas, I. Katakis, "Multi-Label Classification: An Overview," International Journal of Data Warehousing&Mining, vol. 3, No. 1, 2007, pp. 1-13.

TABLE IV. CLASSIFICATION RESULTS OF MULTI-LABEL CLASSIFIERS ON THE YEAST DATA WITH 10-FOLD CV*

Performance measure	FAM	ML-FAM	BoosTexter [6]	ML-*k*NN [6] *k*=8	ML-*k*NN [6] *k*=9	ADT [6]	Rank-SVM [6]	DML-*k*NN [7] *k*=9	BP-MLL [8]
HL	0.229	0.202	0.220	0.195	0.193	0.207	0.207	0.195	0.206
OE	-	0.237	0.278	0.233	0.230	0.244	0.243	0.226	0.233

**A*, *R* and *P* measures were not used in the references.