Fuzzy ARTMAP with Binary Relevance for Multi-label Classification

Lik Xun Yuan[1], Shing Chiang Tan[1](✉), Pey Yun Goh[1],
Chee Peng Lim[2], and Junzo Watada[3]

[1] Faculty of Information Science and Technology,
Multimedia University, Cyberjaya, Malaysia
lxyuan0420@gmail.com, {sctan,pygoh}@mmu.edu.my

[2] Institute for Intelligent Systems Research and Innovation,
Deakin University, Burwood, Australia
chee.lim@deakin.edu.au

[3] Department of Computer and Information Sciences,
Universiti Teknologi PETRONAS, Seri Iskandar, Malaysia
junzo.watada@utp.edu.my

Abstract. In this paper, we propose a modified supervised adaptive resonance theory neural network, namely Fuzzy ARTMAP (FAM), to undertake multi-label data classification tasks. FAM is integrated with the binary relevance (BR) technique to form BR-FAM. The effectiveness of BR-FAM is evaluated using two benchmark multi-label data classification problems. Its results are compared with those other methods in the literature. The performance of BR-FAM is encouraging, which indicate the potential of FAM-based models for handling multi-label data classification tasks.

Keywords: Fuzzy ARTMAP · Binary relevance · Multi-label classification

1 Introduction

Multi-label data classification is different from the traditional single-label classification problems. In the later, each data sample is assigned to a class from a set of predefined class labels, while in the former, each data sample could be labeled with more than one class [1]. The usefulness of multi-label data classification has been demonstrated in several research areas. As an example, in semantic scene classification [2], a home picture can be annotated with at least one conceptual class such as *sofa*, *chair* and *tv monitor* simultaneously. Similarly, in semantic video categorization, a violent video [3] can be annotated as *rope* and *bind* simultaneously. Other applications include social video [4] and music [5] classification into emotions, as well as protein function prediction [6].

Recently, multi-label data classification has attracted close attention by the machine-learning community. Conventional machine learning models can be used for classifying data samples with a single label. To perform multi-label data classification, these machine-learning models need to be modified, e.g. customized *k*-nearest neighbour (*k*NN) [7] and support vector machine (SVM) [8, 9] models.

I. Czarnowski et al. (eds.), *Intelligent Decision Technologies 2017*,
Smart Innovation, Systems and Technologies 73, DOI 10.1007/978-3-319-59424-8_12

In this paper, a supervised artificial neural network based on the adaptive resonance theory (ART) is proposed to classify data samples into multiple classes. Specifically, a fuzzy adaptive resonance theory with mapping (Fuzzy ARTMAP or simply FAM [10]) is integrated with a binary relevance [11, 12] technique to form BR-FAM. The organization of this paper is as follows. In Sect. 2, the state-of-art of multi-label data classification methods is described. In Sect. 3, the methods for designing BR-FAM are explained. In Sect. 4, BR-FAM is evaluated using two benchmark multi-label data sets, with the results compared and analyzed. A summary of this research work is presented in Sect. 5.

2 Literature Review

In general, methods for learning multi-label data samples can be divided into two groups, namely *problem transformation* and *algorithm adaptation* [12]. The methods of *problem transformation* are applied to convert multi-label data samples into at least a set of one single-label data samples, either with or without considering label ranking subject to relevancy of a query of interest. On the other hand, the methods of *algorithm adaptation* are extension from single-label classifiers, and they classify multi-label data samples directly.

Four popular methods of *problem transformation* are binary relevance (BR) [11, 12], label power-set (LP) [12], ranking by pairwise comparison (RPC) [13], and calibrated label ranking (CLR) [14]. BR is a data transformation technique to decompose a multi-label data set into several single-label binary data sets. The idea of BR is described in detail in Sect. 3.1. BR and its variant [15] have been integrated with some base classifiers, which include decision tree [15, 16], Naive Bayes [15], *k*-nearest neighbor [15] and support vector machine [15]. LP converts each unique set of labels of a data sample into a new single label. When a new data instance is provided, the classifier assigns it to a class label that actually indicates a set of labels. The number of transformed labels in LP depends on the total number of class labels in a data set and also the combination of these class labels assigned to the data samples. RPC converts a multi-label data set into binary data sets. Each data set is based on a pair of labels, and consists of data samples of either class label but not both. Each binary data set, RPC is assigned to a classifier for training. Given a new instance, all classifiers in RPC make predictions. The final output is determined by ranking the votes of each class label. CLR is an extended version of RPC. It introduces an artificial label for multi-label ranking. The artificial label is a breaking point between relevant and irrelevant labels.

On the other hand, the learning algorithms of several single-label classification methods have been modified to perform multi-label classification. They are, for instance, multi-label variants of *k*-nearest neighbor [7], decision tree [18], support vector machine [19] and neural network [20, 21] models.

3 Methods

BR-FAM is an extended version of the original FAM model. It is proposed to deal with multi-label data classification tasks. The details of BR-FAM are as follows.

3.1 Binary Relevance (BR)

BR [11, 12] is one of the popular problem transformation techniques [12] dealing with a multi-label data set. The core idea of BR is to divide a multi-label data set into two groups: either relevant or irrelevant to a class label of interest. BR is algorithm independent. It transforms a multi-label data set into at least one single label data set for a classifier to perform supervised learning.

Assume $L = \left\{\lambda_j : j = 1, \cdots, c\right\}$ is a set of labels in a multi-label data set; $D = \{(\boldsymbol{x}_i, Y_i), i = 1, \cdots, m\}$ is a set of original multi-label data samples, where $\boldsymbol{x}_i$ denotes a feature vector, $Y_i \subseteq L$ represents the corresponding multi labels of the i-th sample. BR processes the original data set D into c data sets with two classes D_{λ_j}, $j = 1, \cdots, c$ where all data samples from D having λ_j are labeled positively, otherwise labeled negatively.

3.2 Fuzzy ARTMAP (FAM)

FAM [10] consists of two fuzzy ART modules that are connected through a map field, F^{ab}. One of these two fuzzy ART modules is the input module that processes the input vectors, whereas another is the output module that processes the output labels. Each fuzzy ART model contains nodes interconnected in three layers: (i) a normalization layer, F_0, that normalizes an M-dimensional input vector $\boldsymbol{a}$ or an N-dimensional output label $\boldsymbol{b}$ through a complement-coding process [10] to a 2 M- dimensional input vector $\boldsymbol{A}$ or 2 N-dimensional output vector $\boldsymbol{B}$ (i.e., $\boldsymbol{A} = (\boldsymbol{a}, \boldsymbol{1} - \boldsymbol{a})$ or $\boldsymbol{B} = (\boldsymbol{b}, \boldsymbol{1} - \boldsymbol{b})$); (ii) an input layer that receives $\boldsymbol{A}$ (or $\boldsymbol{B}$); (iii) a recognition layer that contains a group of prototype nodes whereby each prototype node represents a cluster of information elicited from training samples. The map field is an associative memory that links the prototype nodes from the F_2 layer of the input and output fuzzy modules during training. FAM undergoes an incremental learning process wherein new prototype nodes can be added to F_2 to store new information.

Both the input and output modules perform the same information processing operation. After the input vector $\boldsymbol{a}$ is complement-coded to $\boldsymbol{A}$, it is forwarded to F_2^a, where a choice function [10] is utilized to compute the activation of each prototype node with respect to $\boldsymbol{A}$, as follows:

$$T_j = \frac{\left|\boldsymbol{A} \wedge \boldsymbol{w}_j^a\right|}{\alpha + \left|\boldsymbol{w}_j^a\right|} \tag{1}$$

where α is the choice parameter, which is set to a small positive value close to 0 [10]; $\boldsymbol{w}_j^a$ denotes the connection weight of the j-th prototype node; $\wedge$ represents the fuzzy AND operator that performs element-wise minimum of two vectors. The prototype node with the highest activation, namely node J, is identified as the winning node. A vigilance test is applied to compute the similarity between $\boldsymbol{w}_J^a$ and $\boldsymbol{A}$ against a vigilance parameter [10] $\rho_a \in [0, 1]$.

$$\frac{\left|\boldsymbol{A} \wedge \boldsymbol{w}_J^a\right|}{|\boldsymbol{A}|} \geq \rho_a \tag{2}$$

If the vigilance test is not passed, a new cycle of search for the next winning prototype node is undergone. This search process for a new winning prototype node is only terminated once the winning node succeeds to pass in the vigilance test. Nevertheless, when none of the existing prototype nodes can satisfy the vigilance test, a new prototype node is introduced in F_2^a to encode $\boldsymbol{A}$.

After each fuzzy ART module has identified a winning node, a map-field vigilance test [10] is executed to evaluate prediction accuracy, as follows:

$$\frac{\left|\boldsymbol{y}^b \wedge \boldsymbol{w}_J^{ab}\right|}{\left|\boldsymbol{y}^b\right|} \geq \rho_{ab} \tag{3}$$

where $\boldsymbol{y}^b$ denotes the output vector; $\boldsymbol{w}_J^{ab}$ represents the connection weight of the winning node from F_2^a to F^{ab}; and $\rho_{ab} \in [0, 1]$ represents the map-field vigilance parameter.

If the map-field vigilance test fails, it indicates an incorrect prediction of the output class. Consequently, a match-tracking process [10] is triggered, where ρ_a is raised slightly higher from its baseline setting of $\bar{\rho}_a$ as follows:

$$\rho_a = \frac{\left|\boldsymbol{A} \wedge \boldsymbol{w}_J^a\right|}{|\boldsymbol{A}|} + \delta \tag{4}$$

where δ is set as a positive value close to 0. The adjustment of ρ_a causes the vigilance test in the input fuzzy module to fail. As such, a new search cycle in the input fuzzy module is initiated again with the updated ρ_a setting. The effort for searching a winning node is continuously made until a correct prediction of the output class is made.

When the map-field vigilance test is satisfied, a learning process ensues where $\boldsymbol{w}_J^a$ is updated [10] as follows:

$$\boldsymbol{w}_J^{a(new)} = \beta_a\left(\boldsymbol{A} \wedge \boldsymbol{w}_J^{a(old)}\right) + (1 - \beta_a)\boldsymbol{w}_J^{a(old)} \tag{5}$$

where $\beta_a \in [0, 1]$ denotes the learning parameter of the input fuzzy module. The output fuzzy module undergoes the same operation for pattern matching and learning as in the input fuzzy module from Eqs. (1)–(5) by replacing a with b.

3.3 Fuzzy ARTMAP with Binary Relevance (BR-FAM)

BR-FAM is a modified version of FAM for tackling multi-label data classification tasks. In BR-FAM, an L-label data set D ($L = \{\lambda_j : j = 1, \cdots, c\}$) is converted to c datasets. Each D_{λ_j} contains data samples with binary classes subject to a class λ_j of interest. In this case, a total of c FAM models are created. Each FAM is trained with D_{λ_j}. The outputs are the union prediction of λ_j made by all FAMs.

Two performance metrics are used to measure classification performance of BR-FAM. They are from the harmonic mean of precision and recall, namely the F measure [22]:

$$F1 = \frac{2 * tp}{2 * tp + fp + fn} \tag{6}$$

where tp denotes the number of true positive correctly classified; fp denotes the number of false positive; fn denotes the number of false negative. These two performance metrics are micro-averaged and macro-averaged versions of $F1$, i.e., micro $F1$ (B_{micro}) and macro $F1$ (B_{macro}) [23, 24]. For clarity, consider a binary classification task of D_k,

$$B(tp_k, fp_k, tn_k, fn_k) \text{ for } k = 1, \cdots, c \tag{7}$$

where fp_k, fp_k, tn_k, fn_k are respectively the number of true positive, false positive, true negative and false negative after classifying samples from D_k, then

$$B_{micro} = B\left(\sum_{k=1}^{c} tp_k, \sum_{k=1}^{c} fp_k, \sum_{k=1}^{c} tn_k, \sum_{k=1}^{c} fn_k\right) \tag{8}$$

$$B_{macro} = \frac{1}{c} \sum_{k=1}^{c} B(tp_k, fp_k, tn_k, fn_k) \tag{9}$$

4 Evaluation

4.1 Benchmark Data

Two multi-label data sets that are available from Mulan [25] are used in the experiment to evaluate the classification performance of BR-FAM. The *scene* data set comprises numerical records of 2407 images that are labeled up to 6 concepts, for example, *beach*, *field*, and *mountain*. The *yeast* data set contains numerical records of 2417 micro-array expressions and phylogenetic profiles that are labeled with at least one of 14 functional categories such as *metabolism*, *energy*. Table 1 lists the statistics of both data sets in terms of number of instances, input features, and labels.

Table 1. Information of two multi-label data sets

Dataset	Number of instances (#Training: #Test)	Number of input features	Number of labels
Scene	2407 (1211:1196)	294	6
Yeast	2417 (1500:917)	103	14

4.2 Experimental Setup

We refer to the experimental setup as in [23] to execute BR-FAM for ten times with different sequences of training data samples. Upon completion of a training session with all training samples, the classification performance of BR-FAM is evaluated with all test data samples. Each FAM is trained using $\bar{\rho}_a = 0.5$ and $\beta_a = \beta_b = 1$ within ten epochs. The numbers of training and test samples from the two data sets are listed in Table 1, which follows the original quantity of data samples in the training and test sets in [25]. The classification results of BR-FAM are averaged.

4.3 Results and Analysis

The classification performance of BR-FAM is compared with C4.5 integrated with: (i) different problem transformation methods [23], which include BR, LP, Calibrated Label Ranking (CLR) [14] and two efficient versions of LP (i.e., Random *k*-Labelsets of a disjoint version, namely RA*k*EL$_d$, and Random *k*-Labelsets of an overlapping version, namely RA*k*EL$_o$); (ii) two modified methods for multi-label data classification, which include a multi-label version of the backpropagation algorithm for perceptrons (BPMLL) [20] and a multi-label version of *k*-nearest neighbor algorithm (ML*k*NN) [7]. Notably, except for BR-FAM, all the aforementioned classification methods used in this benchmark study had been trained with 66% of the samples from the entire data set and the rest as the test samples [23]. For clarity, BR-FAM has been trained using fewer number of data samples, i.e., approximately 50% of *scene* and 62% of *yeast* data sets. The rationale is to compare rigorously the classification performance between BR-FAM and those of existing multi-label classification methods.

Tables 2 and 3 present the classification results in terms of micro $F1$ (based on B_{micro}) and macro $F1$ (based on B_{macro}) among BR-FAM, the four versions of multi-label C4.5 (with BR, LP, RA*k*EL$_d$, and RA*k*EL$_o$), CLR, ML*k*NN, and BPMLL. From these results, BR-FAM achieves the highest rates of micro $F1$ and macro $F1$ when classifying the *scene* data set. The classification performances of BR-FAM are moderate in *yeast* where its micro $F1$ is ranked at the sixth position and its macro $F1$ is the second highest among the eight classifiers. Based on these results, BR-FAM appears to be a moderate model for multi-label data classification. However, a further analysis of the results of BR-FAM and a group of five multi-label classifiers developed using different problem transformation methods (i.e., CLR and the four C4.5 versions with BR, LP, RA*k*EL$_d$ and RA*k*EL$_o$) in the *yeast* classification task is made. BR-FAM could achieve micro $F1$ (i.e., 55.15%) that is within the performance range of these five classifiers (53.04%–61.89%). On the other hand, BR-FAM is inferior to ML*k*NN and BPMLL. These two multi-label classifiers have been developed by an algorithm

Table 2. The results of micro F1 (standard deviation is typed in round brackets)

Classifier	Classification task (%)	
	Scene	Yeast
BR	62.36 (1.01)	57.67 (1.89)
LP	60.05 (1.14)	53.04 (1.03)
ML*k*NN	72.29 (1.08)	**63.93 (1.06)**
RA*k*EL$_d$	59.87 (0.82)	54.26 (0.58)
RA*k*EL$_o$	69.58 (1.53)	61.89 (0.74)
CLR	62.82 (0.92)	61.69 (1.29)
BPMLL	48.18 (5.19)	63.11 (1.47)
BR-FAM	**77.43** (3.24)	55.15 (0.69)

Table 3. The results of macro F1 (standard deviation is typed in round brackets)

Classifier	Classification task	
	Scene	Yeast
BR	63.41 (0.91)	38.29 (0.59)
LP	61.04 (1.16)	37.26 (1.09)
ML*k*NN	72.63 (1.37)	36.34 (0.79)
RA*k*EL$_d$	60.90 (0.88)	38.84 (0.50)
RA*k*EL$_o$	70.26 (1.64)	40.66 (0.77)
CLR	64.23 (0.89)	38.52 (0.96)
BPMLL	51.29 (5.26)	**42.85 (1.02)**
BR-FAM	**78.58 (4.35)**	41.46 (0.76)

adaptation approach achieving micro F1 within between 63% and 64%. In other words, the performance of BR-FAM in *yeast* is competitive with those classifiers developed using the same approach, i.e., the problem transformation methods.

5 Summary

In this paper, the FAM model is integrate with a binary relevant technique to handle multi-label data classification problems. The effectiveness of BR-FAM is evaluated using two benchmark data sets. The empirical results show that BR-FAM is comparable with other multi-label classifiers, especially those developed with problem transformation approach.

As part of future work, additional experiment will be carried out to evaluate the classification capability of BR-FAM using additional multi-label data sets available in different application areas. We will also develop a multi-label FAM model using the algorithm adaptation approach.

References

1. Boutell, M.R., Luo, J., Shen, X., Brown, C.M.: Learning multi-label scene classification. Pattern Recogn. **37**, 1757–1771 (2004)
2. Tamaazousti, Y., Le Borgne, H., Popescu, A.: Constrained local enhancement of semantic features by content-based sparsity. In: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ICME 2016, pp. 119–126. ACM, New York (2016)
3. Li, X., Huo, Y., Jin, Q., Xu., J.: Detecting violence in video using subclasses. In: Proceedings of the 2016 ACM on Multimedia Conference, MM 2016, pp. 586–590. ACM, Amsterdam (2016)
4. Chávez-Martínez, G., Ruiz-Correa, S., Gatica-Perez, D.: Happy and agreeable?: multi-label classification of impressions in social video. In: Proceedings of the 14th International Conference on Mobile and Ubiquitous Multimedia, MUM 2015, pp. 109–120. ACM, Austria (2015)
5. Lin, Y.-C., Yang, Y.-H., Chen, Homer H.: Exploiting online music tags for music emotion classification. ACM Trans. Multimedia Comput. Commun. Appl. **7 s**, Article 26 (2011)
6. Yu, G., Rangwala, H., Domeniconi, C., Zhang, G., Yu, Z.: Protein Function Prediction with Incomplete Annotations. IEEE/ACM Trans. Comput. Biol. Bioinf. **11**, 579–591 (2014)
7. Zhang, M.-L., Zhou, Z.-H.: ML-kNN: a lazy learning approach to multi-label learning. Pattern Recogn. **40**, 2038–2048 (2007)
8. Xu, J.: An extended one-versus-rest support vector machine for multi-label classification. Neurocomputing **74**, 3114–3124 (2011)
9. Xu, J.: Multi-label core vector machine with a zero label. Pattern Recogn. **47**, 2542–2557 (2014)
10. Carpenter, G.A., Grossberg, S., Markuzon, N., Reynolds, J.H., Rosen, D.B.: Fuzzy artmap: a neural network architecture for incremental supervised learning of analog multidimensional maps. IEEE Trans. Neural Netw. **3**, 698–713 (1992)
11. Godbole, S., Sarawagi, S.: Discriminative methods for multi-labeled classification. Lecture Notes in Artificial Intelligence **3056**, 22–30 (2004)
12. Tsoumakas, G., Katakis, I.: Multi-label classification: an overview. Int. J. Data Warehous. Min. **3**, 1–13 (2007)
13. Hüllermeier, E., Fürnkranz, J., Cheng, W., Bringer, K.: Label ranking by learning pairwise preferences. Artif. Intell. **172**, 1897–1916 (2008)
14. Fürnkranz, J., Hüllermeier, E., Mencia, L., Brinker, K.: Multi-label classification via calibrated label ranking. Mach. Learn. **73**, 133–153 (2008)
15. Cherman, E.A., Monard, M.C., Metz, J.: Multi-label problem transformation methods: a case study. CLEI Electron. J. **14**, 4 (2011)
16. Tanaka, E.A., Nozawa, S.R., Macedo, A.A., Baranauskas, J.A.: A multi-label approach using binary relevance and decision tree applied to functional genomics. J. Biomed. Inf. **53**, 85–95 (2015)
17. Chou, S., Hsu, C.-L.: MMDT: a multi-valued and multi-labeled decision tree classifier for data mining. Expert Syst. Appl. **28**, 799–812 (2005)
18. Wu, Q., Ye, Y., Zhang, H. Chow, Tommy W.S., Ho, S.-S.: ML-TREE: a tree-structure-based approach to multilabel learning. IEEE Trans. Neural Netw. Learn. Syst. **26**, 430–443 (2015)
19. Chen, W.-J., Shao, Y.-H., Li, C.-N., Deng, N.-Y.: MLTSVM: a novel twin support vector machine to multi-label. Learning **52**, 61–74 (2016)
20. Zhang, M.-L., Zhou, Z.-H.: Multi-label neural networks with applications to functional genomics and text categorization. IEEE Trans. Knowl. Data Eng. **18**, 1338–1351 (2006)

21. Chen, Z., Chi, Z., Fu, H., Feng, D.: Multi-instance multi-label image classification: a neural approach. Neurocomputing **99**, 298–306 (2013)
22. van Rijsbergen, C.J.: Information Retrieval. Butterworths, London (1979)
23. Tsoumakas, G., Katakis, I., Vlahavas, I.: Random k-labelsets for multilabel classification. IEEE Trans. Knowl. Data Eng. **23**, 1079–1089 (2011)
24. Yang, Y.: An evaluation of statistical approaches to text categorization. J. Inf. Retrieval **1**, 78–88 (1999)
25. Mulan: a Java library for multi-label learning. http://mulan.sourceforge.net/datasets-mlc.html