2015 IEEE 17th International Conference on High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), and 2015 IEEE 12th International Conf on Embedded Software and Systems (ICESS)

An Improved Classifier Chain Algorithm for Multi-label Classification of Big Data Analysis

Zhilou Yu*[†], Qiao Wang* , Ying Fan[†] , Hongjun Dai[‡][†] and Meikang Qiu [§]
*School of Information Science and Engineering, Southeast University, 210096, Nanjing, China
Email: yuzhl@inspur.com, qiaowang@seu.edu.cn
[†]Technology Center, Inspur Inc., 250101, Jinan, China
Email: fanying@inspur.com
[‡]Department of Computer Science and Technology, Shandong University, 250101, Jinan, China
Email: dahogn@sdu.edu.cn
[§]Department of Computer Science, Pace University, New York, USA
Email: mqiu@pace.edu

***Abstract*—The widely known classifier chains method for multi-label classification, which is based on the *binary relevance* (BR) method, overcomes the disadvantages of BR and achieves higher predictive performance, but still retains important advantages of BR, most importantly low time complexity. Nevertheless, despite its advantages, it is clear that a randomly arranged chain can be poorly ordered. We overcome this issue with a different strategy: Several times K-means algorithms are employed to get the correlations between labels and to confirm the order of binary classifiers. The algorithm ensure the right correlations be transmitted persistently as great as possible by improve the earlier predictions accuracy. The experimental results on the Reuters-21578 text chat data set and image data set show that the approach is efficient and appealing in most cases.**

I. Introduction

In many real-world applications, an instance may belong to several predefined topics. This is called *multi-label learning* (MLL) [1] tasks. For example, in text categorizations, each news in web pages may have several themes, and a patent may belong to several fields at the same time. While, in scene classification, each scene image may belong to several semantic class, such as beach and urban [2]. The multi-label context receives more increased attention, and it is applicable to a wide variety of domains, including text classification, scene classification, video classification, and biomathematics.

Generally, existing strategies to solve MLL can be characterized into two categories: to perform problem transformation, or to modify an algorithm directly to make MLL predictions. First, Problem can be transformed from a multi-label problem into one or more single-label problems. In this way, single-label classifiers are employed; and their single-label predictions are transformed into multi-label predictions. This method is simple, direct and easy, but these common algorithms, such as *binary relevance* (BR) [3], always ignore the correlations among the labels of the training data. Then, a consensus view is to take the label correlations into account during the classification process, and it has turned away from BR to more complex method. *Classifier chains* (CC) [4] is proposed with the basis of the BR method, overcomes the disadvantages of BR and achieves a higher predictive performance. Furthermore, some optimized methods are presented to resolve the order of single classification in CC, such as Enhanced CC, probabilistic CC [5]. Also, there are other well-known approaches for multi-label classification, including AdaBoost, decision trees, ML-KNN[6], *multi-label Learning by Exploiting lAbel Dependency* (LEAD) [7], ML-LOC and multi-label neural network.

The main idea of the CC algorithm is to add the 0/1 label relevance of the whole previous classifiers into a set of training instances, So the information of classified labels can be delivered into the rest classifiers. In this method, the order of classifiers in the chain is very important for the predictions accuracy, and this makes a large influence over the results of the predictions.

So, we present a enhanced CC algorithm with K-means cluster method to confirm the order of binary classifiers in this paper. This algorithm, named as Km-CC algorithm, ensures the right correlations be transmitted persistently massively to improve the accuracy of the earlier predictions. selected samples reserve most of the important information of the original training set. In practice, better sample selection techniques should be able to detect and ignore noisy, modify misleading samples. In Km-CC, the quality of training set affects recognition accuracy, and increases efficiency of next sample selection. The prediction performance of the classifiers and classification accuracy can be increased as a result of sample selection, through the removal of noisy and misleading samples.

II. Related Work

Pattern classification is a method capable of discriminating patterns, it is an approach to supervised learning in pattern recognition. The k-nearest neighbor rule (k-means) is a well-known nonparametric decision rule of pattern classification [8]. Let $D = \{(X_1, \theta_1), ..., (X_n, \theta_n)\}$ be the classified sample set. It is independently and identically distributed according to the distribution $F(X, \theta)$ of (X, θ), where the X_{is} take values in a metric space (X, d) and the θ_{is} take values in the set $\{1, ..., c\}$. A new pattern X is given. It is desired to estimate its label θ by the majority of labels $\theta^{[1]}, ..., \theta^{[k]}$ corresponding to the k-nearest neighbors $X^{[1]}, ..., X^{[k]}$ of $\{X_1, ..., X_n\}$ to X. Thereafter, k-means is investigated extensively.

*Qiao Wang is the corresponding author

978-1-4799-8937-9/15 $31.00 © 2015 IEEE
DOI 10.1109/HPCC-CSS-ICESS.2015.240

The k-means determines the class θ of X associated with the largest number of points among the k-means, formally, this method is computationally expensive. To improve the drawback of k-means, Wilson proposed the editing k-means[2], which can be described as follows: θ_i of each X_i is first the estimated by using the k-means and the data set $\{(X_1, \theta_1), ..., (X_n, \theta_n)\}$ is editing by deleting (X_i, θ_i) whenever θ_i does not coincide with its estimate. Finally, the k-means is used again to estimate θ of X by using the editing data. In the editing k-means, editing the reference set is first performed. Firstly, D is divided into reference set D_1 and testing set D_2. Each sample in the reference set is classified by testing set using the k-means [9] and the editing reference set $D_1^{'}$ formed by eliminating its samples from the reference set, that is, all the misclassified samples are then deleted from the reference set, afterward, any input sample is classified using the k-means and the editing reference set. The editing k-means has yielded many interesting results in many finite sample size problems.

From the practical point of view, it effectively improve the classification accuracy rate and reduce the computation times, which makes it suitable as a preprocessor for classification that are much more complex. Nowadays, based on k-means and the editing k-means [10], many interesting modified classifying algorithm have been proposed to find the possibility k-means for land mine detection using sensor data generated by a ground-penetrating radar system, in which, edge histogram descriptors are used for feature extracting and a possibility k-means for confidence assignment.

III. Algorithm Descriptions

A. Models and Preliminaries

In pattern recognition, the k-nearest neighbors algorithm (k-means) is a method for classifying objects based on closest training examples in the feature space. k-means is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-means algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). The k-means classifier is commonly based on the Euclidean distance between a testing sample and the specified training samples.

In this paper, Let $X = R^d$ as the d-dimensional input space, $L = \{l_1, l_2, ..., l_m\}$ as a finite set of possible labels. Given a multi-label training set $D = \{(x_1, l_1), (x_2, l_2), ..., (x_m, l_m)\}$, where $X_i \epsilon X$ is a feature vector. $X = \{x_{i1}, x_{i2}, ..., x_{id}\}$ and $l_i \epsilon L$ is the set of labels associated with X_i. Then, $l_{ij} = 1$ happens only if label j belongs to X_i; otherwise, it is set as 0. The goal of multi-label leaning is to earn a function $H : X \to 2^L$ from D which maps each unseen example to a set of proper labels.

Essentially, CC is a BR method, and it transforms multi-label problems to one label problems. For example, it transforms the output to $H = \{h_1, h_2, ..., h_m\}$. But it is different from BR, because the attribute space for each binary model is extended with the 0/1 label relevance of all previous classifiers. It forms a classifier chain. The training procedure trains the first classifier for label l_i by the original training set D; then, it augments the attribute space by the training instances' label values; then, it trains a next classifier for the next label. Obviously, the up steps can be looped until all labels' classifier can be trained. In Table I, it illustrates a prediction process with an example.

TABLE I. A PREDICTION PROCESS OF CC ALGORITHM

$h : X \to$	l
$h_1 : [0, 1, 0, 1, 0, 0, 1, 1, 0]$	1
$h_1 : [0, 1, 0, 1, 0, 0, 1, 1, 0, 1]$	0
$h_1 : [0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0]$	0
$h_1 : [0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0]$	1
$h_1 : [0, 1, 0, 1, 0, 0, 1, 1, 0, 1, 0, 0, 1]$	0

The order of the chain itself will normally have an effect on accuracy. There is the possible effect of error propagation along the chain at the classification time, when one (or more) of the first classifiers predict poorly. This has also been noticed by some authors and they present some method to resolve it. But they still have instability. We overcome this issue with a different strategy to confirm the order of the chain through a cluster result.

the k-means is a pure classification algorithm, without sample selection phase. When a new sample is assigned to a certain class, it is necessary to calculate distance of new sample and every exiting sample in training set. However, when $n \to \infty$, the computation is considerable large. In fact, it is not necessary to calculate all the distance of an unlabeled sample and every exiting sample in training set, in order to find its class, but to calculate distance with some representative samples. A solution to this problem is to select the information rich samples to represent the original training set. The following we will introduce several methods to solve the problem.

B. Algorithm Descriptions

Generally, if the accuracy of one cluster for one label is higher than others, its accuracy of the classifier is also higher, just because the feature of this label outstands more. It is also easy to distinguish, so we confirm the order of chain to calculate the accuracy of cluster for each label.

The editing k-nearest neighbors classifier consists of the k-nearest neighbor classifier and an editing training set. In order to reduce the classification error rate and improve the classification accuracy rate, it selects samples from training set which accurately classified as the editing training set (training set). When we classify every sample of testing set, it is evaluated distance of the sample and each pattern in training set and then is assigned to a certain class by k-means. This can save a host of computation time. The procedure of the algorithm can be shown as the following.

First, do cluster on the training set using K-means method with $k = 2$. It will get two sample sets S (sample number denoted n_s) and T (sample number denoted). Then, it can be found that $n_s + n_T = n$.

Second, statistics the number of each label $l_i = 0$ in S and T, to get s_{i0} and t_{i0} respectively. The, $l_i = 1$ in S and T, to get s_{i1} and t_{i1}. The formulas are:

$$s_{i1} = \sum_{w=1}^{n_s} l_{wi} \tag{1}$$

$$s_{i0} = n_s - s_{i1} \tag{2}$$

$$t_{i1} = \sum_{w=1}^{n_t} l_{wi} \tag{3}$$

$$t_{i0} = n_t - t_{i1} \tag{4}$$

In the formulas, $l_{wi}(1 < i < m)$. w is the index of the instance in S or T; i is the index of label number to calculate; l_{wi} indicates the i'th label value (0/1) of w'th instance. By the described above, we can see that the sample number in the training set of the improved method is more than in original method. This is because the number of the training set is rationally increased in the k-means. In the same time it can retain the important information-rich samples that k-means has lost.

Third, calculate the cluster's accuracy c_i of $l_i(0 < i \leq m)$, the formula is:

$$c_i = \frac{(|s_{i0} - t_{i0}| + |s_{i1} - t_{i1}|)}{n} \tag{5}$$

Forth, select the initial cluster center in random, and repeat the upper step (1) to (3) for λ times, then calculate the average $\bar{c_i}$ for all c_i.

Finally, sort descending the labels according to $\bar{c_i}$ and get the result as the CC chain's order.

In order to reduce the loss of information and improve the classification accuracy rate, in this paper, we modify the previous method. The final editing training set is no longer a subset of reference set but an union of two subsets respectively belongs to reference set and testing set. The detailed process is as follows:

Let $D = \{X_1, X_2, ..., X_1\}$ be training set, and each sample has a given class label. Then we divide D into two subsets D_1 and D_2, in which:

$$D_1 = \{Y_1, Y_2, ..., Y_M\} \, (1 \leq M \leq n) \tag{6}$$

$$D_2 = \{Z_1, Z_2, ..., Z_N\} \, (1 \leq N \leq n) \tag{7}$$

$n = M + N$. Y_i and Z_j be training samples, each has p features $\{y_{i1}, y_{i2}, ..., y_{ip}\}, \{z_{j1}, z_{j2}, ..., z_{jp}\}, (i = 1, 2, ..., M; j = 1, 2, ..., N)$ and let D_1 be reference set (training set), D_2 be testing set. Classify every sample belongs to D_2 by D_1 once more. Namely, for every $Z_j \in D_2$ and $Y_i \in D_1$, they are evaluated distance of Z_j and Y_i as follows:

$$D(Z_j, Y_i) = \sqrt{\sum_{r=1}^{p} (Z_{jr} - y_{ir})^2} \tag{8}$$

Then, use K-means to assign class of Z_j, and preserve the results. Estimate whether each sample in D_2 misclassified or not, if misclassified, then remove the corresponding test sample from D_2. Remember the rest samples which are accurately classified in D_2 for $D_2^{'}$:

$$D_2^{'} = \left\{ Z_1^{'}, Z_2^{'}, ..., Z_r^{'} \right\} (1 \leq r \leq N) \tag{9}$$

Serve $D_2^{'}$ as editing training set. For an unlabeled vector X_l, it is classified by $D_2^{'}$ using k-means. From the process of implementation, it is obviously found that computation has been greatly reduced. Besides, the approach increases the classification accuracy and effectively removes the noisy or misleading samples which can be definitely discerned.

IV. Experiments and Results

A. Data Sets

Reuters-21578 text chat data set is a standard text classification data set, which contains the text of 135 categories. This paper selected Reuters-21578 which marked as five largest categories and for multi-label data set as the experimental data sets. We got 583 data sets, 420 training set and 163 testing set. The data sets used in this experiment are two two-dimensional synthetic data sets which are generated as follows: each class has 2000 samples which were independent and identically distributed $(i.i.d)$, drawn from a normal distribution having mean as $(2, 3)$, $(3, 2)$ and the same covariance matrix as $I_{2\times2}$, (i.e., identity matrix).

The image data set consists of 2,000 natural scene images, where a set of labels is artificially assigned to each image. All the possible class labels are desert, mountains, sea, sunset and trees. The number of images belonging to more than one class (e.g. sea+sunset) comprises over 22% of the data set, many combined classes (e.g. mountains+sunset +trees) are extremely rare. On average, each image is associated with 1.24 class labels.

The experiment reduced data set using the genetic algorithm, and then identified and extracted the keyword feature of data set using the data dictionary, and then got classifier chain sequence of the labels using k-means, finally got the classification results from classifier chain sequence.

B. Results

Ten-fold cross-validation is performed on each experimental data set, where Table II, Table III and Table IVreport the detailed results in terms of five evaluation metrics[17]. Where $\triangle$ denotes the smaller the value, the more high performance; also, $\triangle$ denotes the bigger the value, the more high performance; "Bold" indicates the best results

The text chat data set classification results compare with other methods are represented in Table II. The image data set classification results compare with other methods are represented in Table III. The text chat data set classification results with different parameter λ which indicate the cluster times are represented in Table IV. And the Figure 2 given the change trend of different λ in five metrics.

TABLE II. Text Chat Data Set Classification Results

Algorithm	Hamming loss	One-error	Coverage	Ranking loss	Precision
Km-CC	0.173±0.010	0.317±0.030	0.200±0.240	0.208±0.015	0.740±0.021
ECC	0.189±0.013	0.371±0.021	0.219±0.150	0.247±0.022	0.665±0.030
LEAD	0.175±0.021	0.389±0.019	0.208±0.181	0.198±0.017	0.707±0.029
ML-KNN	0.179±0.021	0.383±0.033	0.222±0.312	0.228±0.016	0.661±0.025

TABLE III. Image Data Set Classification Results

Algorithm	Hamming loss	One-error	Coverage	Ranking loss	Precision
Km-CC	0.179±0.012	0.280±0.023	0.204±0.213	0.282±0.017	0.781±0.025
ECC	0.199±0.013	0.351±0.025	0.186±0.220	0.325±0.020	0.695±0.031
LEAD	0.203±0.025	0.282±0.016	0.156±0.141	0.295±0.017	0.715±0.025
ML-KNN	0.216±0.015	0.293±0.014	0.151±0.262	0.278±0.022	0.670±0.028

TABLE IV. Text Chat Data Set Classification Results for different λ

λ	Hamming loss	One-error	Coverage	Ranking loss	Precision
5	0.233±0.034	0.375±0.025	0.304±0.125	0.335±0.035	0.655±0.029
8	0.189±0.013	0.356±0.021	0.279±0.150	0.247±0.022	0.685±0.032
12	0.173±0.010	0.317±0.030	0.200±0.240	0.228±0.019	0.740±0.021
15	0.171±0.021	0.310±0.019	0.208±0.132	0.225±0.027	0.747±0.039
18	0.169±0.021	0.305±0.023	0.206±0.212	0.208±0.018	0.750±0.045
20	0.168±0.015	0.302±0.013	0.203±0.225	0.203±0.025	0.752±0.065

As shown in Table II, for the text chat data set, Km-CC is superior to the compared algorithms in hamming loss, one-error, coverage, precision, especially in one-error and precision. It is a little worse to LEAD in ranking loss.

As shown in Table III, for the image data set, Km-CC is superior to the compared algorithms in hamming loss, one-error, precision, especially in one-error and precision. It is a little worse to ML-KNN in ranking loss. But it is worst to the compared algorithms in coverage.

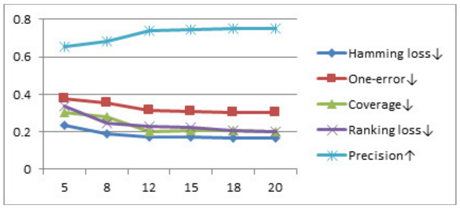

Fig. 1. The influence of λ in different metrics

As shown in Table IV and Figure 1, with the increase of λ value, the parameter values gradually stabilized. Consider the time complexity and accuracy comprehensive, $\lambda = 12$ is the best parameter value. it can be clearly seen that the retention rate of EKNN is only 35.6%. This is mean that only fewer samples have been remained, which may not represent distribution of the whole training set. While Km-CC is employed, the retention rate rises to 72.25%. This is mean that while removes the noise or misleading samples, retains as much as possible the information-rich samples. In other words, Km-CC can compensate for the shortcoming of Km-CC to make classification accuracy higher.

V. Conclusions

In this paper, a novel approach to multi-label learning is proposed by exploiting the dependencies among labels. Specifically, K-means algorithm is employed to get the correlations between labels and to confirm the order of binary classifiers. The algorithm ensure the right correlations be transmitted persistently as great as possible to improve the earlier predictions accuracy. The experimental results on the Reuters-21578 text chat data set and image data set show that the Km-CC approach is efficient and appealing in most cases.

References

[1] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 8, pp. 1819–1837, Aug 2014.

[2] J. Jiang and L. McQuay, "Predicting protein function by multi-label correlated semi-supervised learning," *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, vol. 9, no. 4, pp. 1059–1069, July 2012.

[3] D. Tao, X. Li, and S. Maybank, "Negative samples analysis in relevance feedback," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 19, no. 4, pp. 568–580, April 2007.

[4] J. Read, B. Pfahringer, G. Holmes, and E. Frank, "Classifier chains for multi-label classification," *Machine learning*, vol. 85, no. 3, pp. 333–359, 2011.

[5] A. Kumar, S. Vembu, A. K. Menon, and C. Elkan, "Learning and inference in probabilistic classifier chains with beam search," in *Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 665–680.

[6] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[7] M.-L. Zhang and K. Zhang, "Multi-label learning by exploiting label dependency," in *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '10. New York, NY, USA: ACM, 2010, pp. 999–1008. [Online]. Available: http://doi.acm.org/10.1145/1835804.1835930

[8] J. C. Bezdek, S. K. Chuah, and D. Leep, "Generalized k-nearest neighbor rules," *Fuzzy Sets and Systems*, vol. 18, no. 3, pp. 237–256, 1986.

[9] M. Kusner, S. Tyree, K. Q. Weinberger, and K. Agrawal, "Stochastic neighbor compression," in *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, 2014, pp. 622–630.

[10] Y. Jing, L. Hu, W.-S. Ku, and C. Shahabi, "Authentication of k nearest neighbor query on road networks," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 26, no. 6, pp. 1494–1506, 2014.